HomologyBasis: Fast Computation of Persistent Homology

Peter Hannagan Brosten

A Master's Thesis in Topology

Department of Mathematics
University of Bergen

June, 2021

Acknowledgements

This thesis serves as the culmination of my two journey to obtain a Master's degree. It has been anything but easy and I am forever indebted to all those who lightened my load along the way. This section is for them.

To my mother, father, and sister. Thank you for supporting me when I was low, cheering me on when I was high, and finding the time to call regardless of the time difference.

To my oldest friend Joe. Thank you for always being willing to stay up late and to listen to me ramble. You have been a constant source of joy and support and I am so thankful our bond is stronger than the ocean is wide.

To my mentors, Morten Brun and Alexander Schmeding. You have my sincerest gratitude for the patience and guidance you gave me these past two years. You each made this department feel like home for a student searching for his place in the world.

This thesis would not be what it is without the help of Tanner Rosenberg and Erlend Raa Vågset for looking over and correcting the rough drafts of this paper. Your insights were invaluable and very much needed.

Lastly, to my friend and mathematician in arms Kristian André Jakobsen. Your impact on my life can never be overstated. You welcomed me to this foreign land, taught me its language and culture, and supported me through some of the hardest times in my life. From the bottom of my heart, thank you for everything you are.

Abstract

Simplicial complexes are used in topological data analysis (TDA) to extract topological features of the data. The HomologyBasis algorithm is proposed as an efficient method for the computation of the topological features of a finite filtered simplicial complex. We build up the implementation and intuition of this algorithm from its theoretical foundation ensuring this schema produces the desired simplicial homlogy groups as claimed. HomlogyBasis implemented and compared with the GUHDI algorithm to determine the HomologyBasis' efficiency at computing persistence pairs for finite filtered simplicial complexes. We find the HomologyBasis algorithm performs much better than GUHDI on large low-dimensional simplicial complexes but needs further refinement before it can more efficiently work with high-dimensional complexes.

Contents

1 Introduction

Data analysis seeks to understand the underlying relationships of a given data set. These relationships can be thought of as defining some sampling manifold upon which all data points lie. In order to understand the relationships, it is prudent to understand the sampling manifold from which a given data set is extracted. This becomes increasingly difficult as the complexity of the data sets increases. There are two usual causes for this complexity. First, the data may be of such high dimension that it renders intuitive visualization of its sampling manifold impossible. Second, as data collection can be imprecise, random fluctuations and disturbances, i.e. noise, in our data set can obfuscate the underlying structure of the manifold.

Topological data analysis (TDA) is a recent framework that seeks to extract this structural information from a metric data set by constructing triangulations of the point cloud and then describing the "shape" of this topological space through the lens of homology. Specifically, homology gives us a qualitative description of the space's structure by characterizing the holes of the space. Persistent homology, introduced by Edelsbrunner in [9], is an extension of homology to data analysis in the attempt to define, recognize, and ignore noise in collections of data. It attempts to siphon out the homological features that are most integral to the topological space from those that are created by the noise within the data. Persistent homology has had many successful applications. In medical research, persistent homology has been used to help in the identification of breast cancer subgroups [11] and liver lesions [1]. In Molecular Dynamics it has been use to assist with model selection [10] as well as being applied to the study of viral evolution [5].

A popular choice of algorithm for computing persistent homology is GUHDI (Geometry Understanding in Higher Dimensions), which utilizes a special class of trees called SimplexTrees to efficiently store simplicial complexes and compute their persistent homology groups. The central problem with computing persistent homology is that the computation time scales exponentially with the number of simplices in the simplicial complex. This leads to a restriction on the data sets that are feasible to extract homological information from. When dealing with "big" data, we quickly encounter computation times so large that any information we hope to gleam from our analysis becomes impractical to extract. Trying to better understand GUHDI, we noticed that it fails to take advantage of the algebraic structures called chain complexes that are so integral to the theory of homology. This leads to the question: can we use chain complexes to further improve upon the GUHDI algorithm?

In this thesis, we present an algorithm for the computation of a finite filtered simplicial complex's persistence pairs. We introduce a variation of the chain com-

plex and demonstrate that these new totally filtered chain complexes can be used to uniquely represent sets of simplicial complexes that share a common ordering and structure. There exists a set of projections which restrict a given totally filtered chain complex to a free k-vector space composed strictly of basis elements that generate homological features. We use these projections to find isomorphisms between each homology group and a unique subspace of our homology basis. Our aptly named algorithm, HomologyBasis, produces a given finite filtered simplicial complex's persistence pairs by converting it to a totally filtered chain complex, building up the subsequent homology bases corresponding to the restriction of our chain complex to the first n basis elements, and tracking when various basis elements appear and vanish from the image of our projections. We prove structuring the HomologyBasis in this way does produce both the desired homology bases and the persistent pairs for all homological features that exist within this sequence. While our implementation of this algorithm is yet to be optimized, initial comparisons with GUHDI highlight the potential for HomologyBasis to compute persistent homology of finite filtered simplicial complexes with great efficiency.

A github repository for the HomologyBasis algorithm has been created and made public here: https://github.com/Pbrosten/HomologyBasis.

Overview

This thesis is structured as follows:

Section 2 introduces the basic concepts used as a mathematical foundation throughout this thesis. Specifically, abstract simplicial complexes, Euclidian data complexes and filtrations, and both simplicial and persistent homology. Section 2.5 serves to introduce the totally filtered chain complex with which we work for the remainder of the thesis.

Section 3 presents homology basis of a totally filtered chain complex. It is shown to contain exactly the homological information introduced in Section 2.3 and a schema for its extraction using compatible sequences of homology bases is described. Expanding upon these compatible sequences, we find they also define the persistent homology form Section 2.4 for the context of totally filtered chain complexes. The proofs of Section 3 are predominantly constructive and thus serve as a foundation for the computation of persistence pairs using totally filtered chain complexes and homology bases.

Section 4 combines the mathematics presented in Sections 2 and 3 and proposes a general construction of the HomologyBaisis algorithm and its formalization as pseudo code. The main difference between GUHDI and the proposed HomologyBasis is that the former explicitly makes use of the chain complex structure that is present when computing homology groups. The hypothesis that this will allow for more efficient computation of these groups is tested and the results are presented in Section 6

Section 5 presents a brief description of the Simplex Tree data structure utilized by GUHDI. This section is for those interested in the structure GUHDI utilizes. However, as the Simplex Tree structure has already been documented in depth by Boissonnat and Maria in [3], only an overview given here.

Section 6 contains the results and analysis of various comparative tests run between HomologyBasis and the GUHDI algorithm. These comparisons us two standard benchmark data sets from [12] in addition to multiple randomly generated point clouds of various dimension.

Sections 7 and 8 serve as the final thoughts and remarks on the HomologyBasis algorithm and its underlying theory. The former focuses on the various problems encountered during the comparisons and elaborates on the areas that require further investigation. The latter gives a final summary of the mechanisms of the HomologyBasis and the results from its comparisons with GUHDI.

2 Preliminaries

The information in this section forms a foundation upon which this thesis is built. The topics pertain predominantly to algebraic and computational topology. Most definitions are drawn from various well regarded texts on these subjects, see [7] and [9]. The reader is expected to have a general understanding of abstract algebra. Readers well versed in the subjects of simplicial methods, homology and computational topology may skip ahead to Section 2.5 and simply refer back to the prior sections when needed.

The three most important topological objects for this thesis are the *simplicial complex*, its *homology groups* (specifically *persistent homology*), and the introduced algebraic structure we have named the *totally filtered chain complex*. We begin by introducing the simplicial complex and developing some of the most important features for its use, specifically applications to data analysis. We discuss the problem of choosing an appropriate simplicial complex for a given data set and what a *filtration* of a simplicial complex is. This is followed by a discussion of the sequences of algebraic groups, referred to as *chain complexes*, and how they allow for the extrapolation of homological information. Finally, we finish our background information with the introduction of an algebraic structure we utilise to more efficiently compute homology (persistent homology) for a given finite filtered simplicial complex: the totally filtered chain complex.

2.1 Simplicial Complexes

There are two types of simplicial complexes: *geometric simplicial complexes* and *abstract simplicial complexes*. The former are topological spaces that are intuitive but difficult to efficiently use in computations. The latter are simplified to only contain pertinent combinatorial information and hence are much more flexible for computational purposes. Note the terms abstract simplicial complex and simplicial complex are used interchangeably for the following discussions.

Definition 2.1. The pair (X, V) of a vertex set V and X a finite nonempty subsets of a vertex set V is an *abstract simplicial complex* if for any $\sigma \in X$ and $\tau \subseteq \sigma$, $\tau \in X$.

An element of X is referred to as a *simplex* or more specifically a *d-simplex* when the element is a finite subset with cardinality $d+1$. If $\tau \subseteq \sigma$ then τ is a *face* of σ and σ is a *coface* of τ. When $\tau \subset \sigma$, τ is called a *proper face* of σ and the *boundary* of a simplex is the union of all its proper faces. If $Y \subseteq X$ and (Y, V) is a simplicial complex, then (Y, V) is a *simplicial subcomplex* of (X, V). When there is no ambiguity about the choice of vertex set an abstract simplical complex

is a based on, we may simply refer to (X, V) as X. When the highest dimensional simplex in X is a k-simplex, we refer to (X, V) as a k-simplicial complex. The simplicial subcomplexes that restrict a k-simplicial complex to all simplices of dimension $l < k$ and lower is the k-simplicial complex's l-skeleton.

Example 2.2. Let the set $V = \{0, 1, 2, 3, 4, 5\}$. Now the collection of subsets

$$X = \{0, 1, 2, 3, 4, 5, 01, 12, 13, 23, 24, 25, 34, 35, 45, 123, 245, 345\}$$

such that each string $v_0v_1...$ represents the simplex $\{v_0, v_1, ...\} \subset V$ is a simplicial complex. In order to get a geometric sense of what this means we may identify each element of V with a point in $\mathbb{R}^n$ for some n and then identifying every non-singleton subset of V with the convex hull defined by the points in $\mathbb{R}^n$ that are identified with the vertices of our subset. We can see an embedding of (X, V) in $\mathbb{R}^2$ in Figure 1 along with its 1-skeleton.

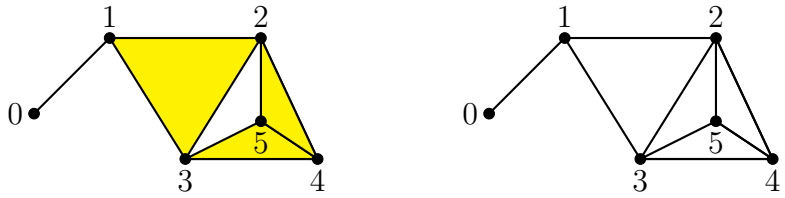


Figure 1: Representation of the simplicial complex and 1-skeleton from Example 2.2, embedded in the plane.

Notice that the representation in Figure 1 is composed of multiple variations of the same type of geometric objects for each dimension. Namely, points for dimension 0, line segments for dimension 1, triangles for dimension 2. Each of these is an embedding of what is referred to as the *standard d-simplex* for some dimension d. These generic building blocks are useful for representing the combinatorial information encoded in an abstract simplicial complex geometrically.

Definition 2.3. The *standard d-simplex*, Δ^d, is the subspace spanned by the unit coordinate vectors of $\mathbb{R}^{d+1}$.

Let us consider the example where $d = 2$. The standard 2-simplex will be the the subset of $\mathbb{R}^3$ that is spanned by the unit coordinate vectors $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$ as seen in Figure 2. Using this new vocabulary, we can aptly describe Figure 1 as being as an embedding of a collection of standard d-simplices for $d = 0$, 1, and 2 such that the embedding preserves the combinatorial information encoded in the abstract simplicial complex from example 2.2. We refer to such an embedding as the *geometric realization* of our abstract simplicial complex (X, V).

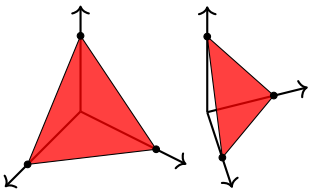


Figure 2: Two views of the standard 2-simplex, Δ^2, constructed using the unit coordinate vectors of $\mathbb{R}^3$.

Definition 2.4. Given a finite simplicial complex (X, V) with n elements in its vertex set. The *geometric realization of* X, denoted $|X|$, is a subspace of $\mathbb{R}^{n+1}$ defined by the embedding of various Δ^d in such a way that each $i \in V$ can be mapped to a unique point $x_i \in \mathbb{R}^{n+1}$.

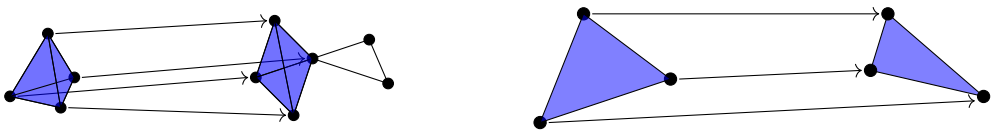
Interestingly, we may always construct a geometric realization for a simplicial complex given we choose a sufficiently high dimensional euclidean space to embed in.

Theorem 2.5 (Geometric Realization Theorem)**.** *Every abstract simplicial complex of dimension d has a geometric realization in $\mathbb{R}^{2d+1}$.*

A proof of the Geometric Realization Theorem is presented by Edelsbrunner in [8].

Definition 2.6. A map $f\colon (X, V) \to (Y, W)$ between simplicial complexes is *simplicial* if each simplex in X is taken to a simplex in Y via a linear map taking vertices to vertices.

It is useful to notice that simplicial maps are purely determined by the restriction to the vertex sets $f^0\colon V \to W$. That is when considering a simplicial map, it is sufficient to only consider what happens to 0-simplices of X under the simplicial map f. Examples of such maps are the inclusion of a 3-simplex in a simplicial complex of degree 3 or the inclusion of a 2-simplex based on one vertex set to another as seen below.



(a) (b)

Figure 3: (a) An inclusion simplicial map of a 3-simplex into a 3-simplicial complex. (b) A simplicial map between two 2-simplices with different vertex sets.

Notice that the simplicial map defined on the simplicial complexes in Figure 3b has a special relation that the other map lacks. The second simplicial map defines an isomorphism between the two simplicial complexes X and Y where as the first is non-isomorphic. In terms of topological features, both simplicial complexes in the second example are identical even though $(X, V) \neq (Y, W)$. This leads us to a useful construction that eliminates the need to explicitly state a vertex set when discussing a simplicial complex. That is, we make equivalent all complexes which have an simplicial isomorphism between them and thus share the same combinatorial and topological information.

Definition 2.7. Let (X, V) and (Y, W) be two simplicial complexes such that there exists a bijective simplicial map $f^0 \colon V \to W$ that induces an isomorphism $f \colon (X, V) \to (Y, V)$. Then $(X, V) \sim (Y, V)$ and both are members of the same equivalence class $[X]$. The equivalence class $[X]$ is referred to as an *isomorphism class of simplicial complexes*.

Using these isomorphism classes, we may construct a generalized collection of abstract simplicial complexes that no longer require any specification of vertex set.

Definition 2.8. Let X^* be the collection of all possible isomorphism classes of simplicial complexes. That is,

$$X^* = \{[X] | \, (X, V) \text{ is an abstract simplicial complex for some vertex set } V\}$$

The set X^* now contains a representative for every possible abstract simplicial complex. As our goal is to apply topoligical data analysis to finite data sets, we limit X^* to strictly finite abstract simplicial complexes. The prior definitions will become useful later in Section 2.5.

2.2 Complexes, Filtrations, and Data

Consider a discrete subset $P \subseteq \mathbb{R}^n$. If P is a finite collection of data points, a *point cloud*, sampled from some unknown manifold, it may be productive to extrapolate information about the underlying manifold in order to better understand the specific data set. Constructing graphs can help to discern pertinent clues about the underlying structure. However, it is usually more descriptive to construct a simplicial complex instead. However, given the numerous different simplicial complexes that can be constructed on any sufficiently large point cloud, choosing a useful simplicial complex for our data set is easier said than done.

When analyzing data, it is necessary to consider what underlying information about the point cloud we are attempting to extract. Hence, it is pertinent to choose a simplicial complex that will help uncover interesting relations within our

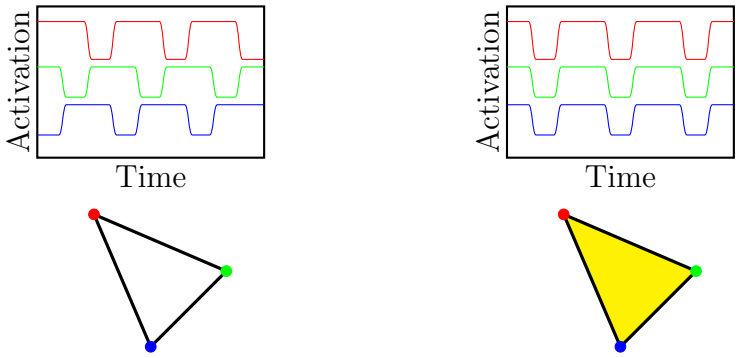


Figure 4: On the left we have the simplicial representation of the pairwise activation neural system. Simplices are added to the complex based on simultaneous activation. On the right, we have same neural system with the addition of triple simultaneous activation represented by the inclusion of a 2-simplex.

data set that may not have been immediately obvious. One common method is to abstract the idea of "nearness". Simplicial complexes can help define complex relations between the specific data points in order to uncover a better sense of the underlying "nearness" of the point cloud. Take the example presented in [6]. Consider a simple three neuron system, with two distinct activation patterns. In the first, each neuron is pairwise-active. That is, at any given time exactly two of the neurons are firing, never three. For the second, let each of the neurons fire in unison. Using the tools of graph theory, an intuitive representation of these systems might to identify each neuron as a vertex and construct a C_3 graph, where every edge between two vertices is added, on our vertex set. Here we use concurrent activation of two neurons to determine when it is appropriate to add an edge. However, this representation falters to differentiate between the two. This is where the use of simplicial complexes becomes advantageous. Choosing our definition of when vertices are "near" one another to be when they are simultaneously active, we are able to differentiate between the two systems by treating the C_3 graphs as simplicial complexes and then filling in the 2-simplex for the latter system, see Figure 4, as all three vertices are "near" one another. This allows us to represent these two systems without losing information about the internal relationships of neuron activation activation patterns. We call such abstract simplicial complexes that have a Eucidean data set as their vertex set a *Euclidean data complex*[1].

[1]This terminology is credited to Morten Brun and Kristian André Jakobsen.

2.2.1 Nerves

When given a data set, one approach to extracting topological information about the sampling manifold utilizes a structure called a *nerve*. The main premise is that when given a covering, we can simplify the cover so that it only preserves the most important information about the underlying structure of the manifold of interest.

Definition 2.9. Consider a finite collection of sets F. The *nerve* of F, $\mathrm{Nrv}(F)$, is defined as:

$$\mathrm{Nrv}(F) = \{X \subseteq F \mid \bigcap X \neq \emptyset\}.$$

That is, the set of all non-empty subcollections whose sets have a non-empty intersection.

A useful property of the nerve is that it is always an abstract simplicial complex, regardless of the collection F. Take the collection of sets in Figure 5. We may think of the nerve of this collection of sets as an abstract simplicial complex by identifying each set in F to a unique vertex in some vertex set of cardinality 4. Then for every subcollection of n sets that share a non-empty multi-intersection we add an $(n-1)$-simplex, defined by vertices corresponding to each set in the subcollection, to our simplicial complex. The nerve of F is isomorphic to the restriction of the standard 3-simplex to is own boundary. Notice in this example, there is a difference between the homotopy types of the union of the F and the nerve of F. This occurs because we are dealing with non-convex sets. Interestingly, if the sets in F are convex then the nerve preserves homotopy type. This is stated formally in the following theorem.

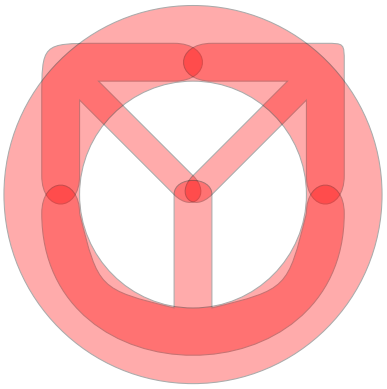


Figure 5: A finite collection of sets F. Notice that viewing the $\mathrm{Nrv}(F)$ as a simplicial complex we get the standard 3-simplex restricted to its boundary.

Theorem 2.10. *Let F be a finite collection of closed, convex sets in Euclidean space. Then the nerve of F and the union of the sets in F have the same homotopy type.*

This theorem becomes very useful when trying to tackle the Euclidean data complex choice problem for a point cloud. In order to extract some interesting topological information about the underlying manifold our data set has been sampled from, we may construct a nerve on some finite collection of closed, convex sets covering our point cloud.

2.2.2 The Čech Complex

Letting the closed, convex sets be d-balls of some given radius, we arrive at a schema for constructing the *Čech complex.*

Definition 2.11. Let S be a finite set of points in $\mathbb{R}^d$ and write $B_r(v) = v + r\mathbb{B}^d$ as the closed d-ball with radius r and center x. The *Čech complex* $\check{\mathrm{C}}_r(S)$ is the simplicial complex of all subsets $\sigma \subseteq S$ where the intersection of all $B_r(v)$ for $v \in \sigma$ is nonempty. That is,

$$\check{\mathrm{C}}_r(S) = \{\sigma \subseteq S \mid \bigcap_{v\in\sigma} B_r(v) \neq \emptyset\}.$$

By construction, the Čech complex is identical to the nerve of this collection of specified d-balls.

Notice that for any collection of vertices S, there are more than one Čech complex for S. By varying the parameter r we can construct a dynamic family of Euclidean data complexes constructed on the vertex set S and bounded by the simplicial complexes $\check{\mathrm{C}}_0(S) = S$ and $\check{\mathrm{C}}_\infty(S)$, where $\check{\mathrm{C}}_\infty(S)$ is the complete $(\|S\|-1)$-simplex. By Theorem 2.10, each member of $\{\check{\mathrm{C}}_r(S)\}_{r=0}^{\infty}$ is homotopic to the union of the radius-r balls about S. When it comes to fidelity of the topological information extracted, the Čech complex is the best choice.

However, this fidelity comes at a two-fold cost. First, the construction of a Čech complex requires that all higher-order intersections must be computed. While this does not present an issue for very small point clouds like the one in Figure 6, it becomes computationally unwieldy when considering larger and larger data sets. Additionally, we are required to store every simplex in our Čech complex individually. Both of these drawbacks have lead to the development of complexes that trade topological fidelity for computational efficiency. Enter the *Vietoris-Rips complex.*

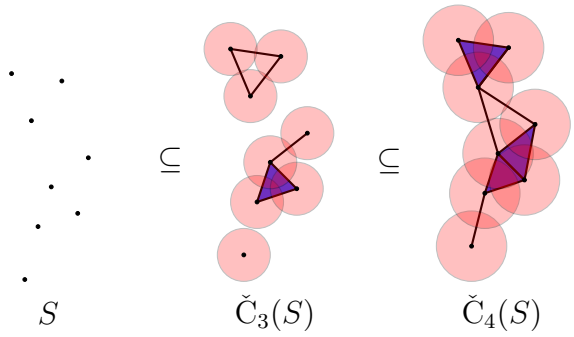


Figure 6: Progression of a Čech complex. The first figure is the point set S in $\mathbb{R}^2$, the second shows the complex $\check{\mathrm{C}}_3(S)$ and the third is the $\check{\mathrm{C}}_4(S)$ complex.

2.2.3 The Vietoris-Rips Complex

Recall for a set of points, the diameter of the set is the maximum of the pairwise distances of all points in that set.

$$\mathrm{diam}(\sigma) = \max(\{\mathrm{dist}(x, y) \mid x, y \in \sigma\}).$$

Definition 2.12. Let S be a finite set of points in $\mathbb{R}^d$. Then given $r \in \mathbb{R}$ the *Vietoris-Rips complex* on S is defined by choosing all subsets of S with diameter less than or equal to $2r$, i.e.,

$$\mathrm{VR}_r(S) = \{\sigma \subseteq S \mid \mathrm{diam}(\sigma) \leq 2r\}.$$

The first difference that should be noticed between the Čech and Vietoris-Rips complexes is when higher dimensional simplices are included. As inclusion is no longer contingent on non-empty multi-intersections, but instead determined only by the largest pairwise distance between vertices in a subset of the point cloud, we find a much more liberal inclusion of higher dimensional simplices in our Vietoris-Rips complex. Take the point cloud S from our discussion of the Čech complex and consider the upper most three points. Remember that in $\check{\mathrm{C}}_3(S)$ we only include the boundary of the 2-simplex defined by these three points. However, when constructing the $\mathrm{VR}_3(S)$ complex, we find that the 2-simplex is included. This can be seen in Figure 7.

This choice to ignore evaluating multi-intersection not only decreases the computational requirements of the Vietoris-Rips complex, but also assists with the issue of efficient storage. Given that the inclusion of higher dimensional simplices is dictated strictly by the pairwise distances of vertices, we find that every Vietoris-Rips complex has a unique 1-skeleton. Utilizing this fact allows us to store any Vietoris-Rips complex by its 1-skeleton. When we need to reconstruct the complex, we may simply add a k-simplex for every complete k-subgraph in the skeleton.

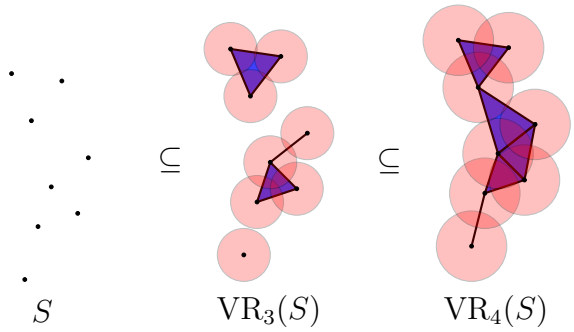


Figure 7: Progression of a Vietoris-Rips complex. The first figure is the point set S in $\mathbb{R}^2$, the second shows the complex $\mathrm{VR}_3(S)$ and the third is the $\mathrm{VR}_4(S)$ complex.

Unfortunately, this increase in efficiency comes at a decrease in topological precision. While $\check{\mathrm{C}}_r(S) \subseteq \mathrm{VR}_r(S)$ for all r and S, the Vietoris-Rips complex only approximates the topological information that can be gleamed from the Čech complex. This is usually considered a good enough approximation for capturing the topology of large-scale holes [4]. Also note that as r approaches ∞, the Čech and Vietoris-Rips complexes converge to the same Euclidean data complex. That is, $\check{\mathrm{C}}_\infty(S) = \mathrm{VR}_\infty(S) = (\mathcal{P}^S, S)$ where S is the data set and $\mathcal{P}^S$ is its powerset.

2.2.4 The Delaunay and Alpha Complexes

Sometimes we are not interested in the higher dimensional relationships that can be expressed by a simplicial complex. In this case it is beneficial to consider either the *Delaunay complex* or its parameterized counterpart, the *Alpha complex*.

When constructing either complex on a point set in $\mathbb{R}^d$, one must first find a cover of $\mathbb{R}^d$ based on the given point set and made up of closed subsets called *Voronoi cells*.

Definition 2.13. Given a finite set of points in $S \subseteq \mathbb{R}^d$, the *Voronoi cell* of a point $u \in S$ is the closed set

$$V_u = \{x \in \mathbb{R}^d \mid \|x - u\| \leq \|x - v\|, \forall v \in S\}.$$

That is, the set of all points in the V_u are as close to the point u as to any other $v \in S$.

An simple example of a Voronoi cell is given in Figure 8. Notice that the Voronoi cells do in fact form a cover $\{V_s\}_{s \in S}$ of $\mathbb{R}^2$. When the collection of Voronoi cells are considered all together, as a cover of $\mathbb{R}^d$, we call them the *Voronoi diagram* of our point set. Having defined the Voronoi diagram, we may now construct the Delaunay complex.

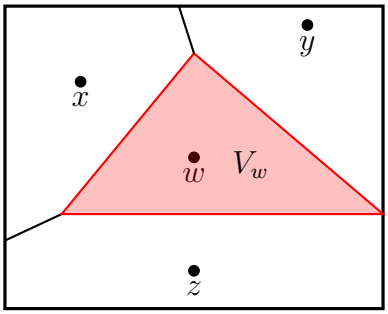


Figure 8: Example of a Voronoi diagram for the point set $S = \{w, x, y, z\}$ in $\mathbb{R}^2$. The Voronoi cell V_w has been highlighted in red. Note that, in this example, the other three Voronoi cells (whose boundaries are in black) will extend indefinitely across the plane and form a cover for $\mathbb{R}^2$.

Definition 2.14. Let S be a finite set of points in $\mathbb{R}^d$. The *Delaunay complex* is isomorphic to the nerve of the Voronoi diagram of S

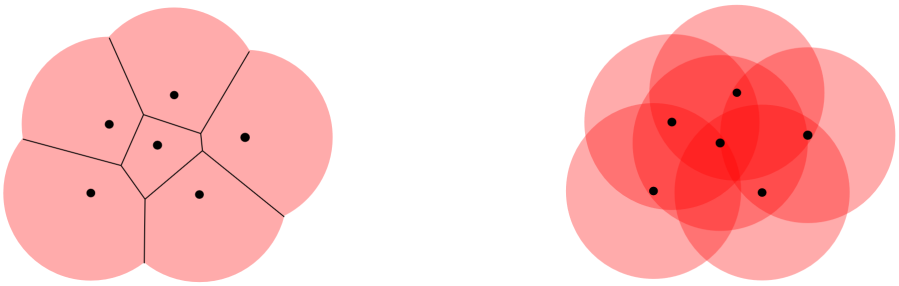
$$\text{Del}(S) = \{\sigma \subseteq S \mid \bigcap_{u \in \sigma} V_u \neq \emptyset\}.$$

The Delaunay complex is very rigid when compared to the Čech and Vietoris-Rips complexes, in that there is only one $\text{Del}(S)$ complex for a given point cloud S. If one desires the malleability of the Čech complex but the higher dimensional suppression of the Delaunay complex, then one can take the parameterization of the latter, which acts as a Čech complex that is bounded by the Delaunay complex. This parameterization is named the *Alpha complex*.

Definition 2.15. Let $V_u^r = V_u \cap B_r(u)$ be a parameterized Voronoi cell. Then given a finite point cloud $S \subseteq \mathbb{R}^d$ and $r \geq 0$, the *Alpha complex* of parameter r is the Euclidean data complex

$$\text{Alpha}_r(S) = \{\sigma \subseteq S \mid \bigcap_{u \in \sigma} V_u^r \neq \emptyset\}.$$

For small values of r, $\text{Alpha}_r(S)$ will be exactly $\check{\text{C}}_r(S)$. However, as r increases and we begin having more and more multi-intersections of d-balls, the bounding by the Voronoi cells start to take effect to limit the number of intersections we must check. This is especially noticeable when there are pockets of densely packed points, as in Figure 9. Having chosen a sufficiently large value of r, we can see that the Čech complex has many multi-intersections which force the inclusion of equally numerous higher dimensional simplices up to dimension 5. In contrast, the bounding effect of the Alpha complex suppresses all d-simplices for $d \geq 3$ in the Čech complex. Thus limiting our simplicial complex to a 3-complex.



(a) $\text{Alpha}_r(S)$ (b) $\check{\text{C}}_r(S)$

Figure 9: Instances of the Alpha and Čech complex for the same value of r on the same densely packed point set. Notice the Alpha complex limits intersections to the boundaries of the Voronoi cells.

2.2.5 Filtrations

The Alpha, Čech, and Vietoris-Rips complexes all share an interesting commonality: when allowing r to vary from 0 to ∞, they each define a family of nested simplicial complexes. To demonstrate why this is of interest to us, consider $\check{\text{C}}_{r*}(S)$ for some $r*$ and S. When viewed in isolation, we can gleam no information about the order in which its simplices were included to built up $\check{\text{C}}_{r*}(S)$. However, when $\check{\text{C}}_{r*}(S)$ is considered as a member of the family $\{\check{\text{C}}_r(S)\}_{r=0}^{\infty}$, we may look back on $r \leq r*$ to better understand how $\check{\text{C}}_{r*}(S)$ was constructed. For each simplex $\sigma \in \check{\text{C}}_{r*}(S)$, assign the value l such that $\sigma \in \check{\text{C}}_l(S)$ and $\sigma \notin \check{\text{C}}_r(S)$ for all $r < l$. Ordering our simplices by these l values will give insight into the way in which our complex $\check{\text{C}}_{r*}(S)$ was built from the point set S.

What we have just done is construct a *filtration* on the simplicial complex $\check{\text{C}}_{r*}(S)$. Intuitively speaking, a filtration gives us a sense of the "time" that each simplex was added to our simplicial complex during its construction. We now formalize this notion.

Definition 2.16. Let X be a simplicial complex. A function $F\colon X \to \mathbb{R}$ is a *filtration* on X if it is non-decreasing with regards to faces of simplices. That is, if τ is a face of σ then $F(\tau) \leq F(\sigma)$.

Each of the parameterized complexes discussed so far has an intuitive filtration we may associate with it. Namely, using the r parameter to define our filtration values. However, the definition is not overly restrictive on what can and cannot be a filtration. For example, given a simplicial complex X, we are allowed to define the filtration function on X that takes all simplices to 6. In this case, the filtration would simply be telling us that the complex X appeared, wholly formed, at the "time" 6. When the only desired feature of the filtration is to demonstrate how

one would build the simplicial complex using one simplex at a time, the implicit *index filtration* becomes the best option.

Definition 2.17. Let X be a finite simplicial complex such that there are $n+1$ simplices in X. Then the *index filtration* on X is the injective function $F\colon X \to \mathbb{N}$ such that the image of F is the totally ordered set $[n] = \{0, 1, \ldots, n\}$.

As there are some many possible filtrations for a given simplicial complex, it is useful to be explicit about both the complex and filtration we are dealing with.

Definition 2.18. Given a simplicial complex (X, V) and $F\colon X \to \mathbb{R}$ a filtration, a *filtered simplicial complex* (X, F) is the application of the filtration to the simplicial complex.

Applying the concept of isomorphism classes of simplicial complexes from definition 2.7, we may decouple a filtered simplicial complexes from its implicit vertex sets.

Definition 2.19. Let the (X, F) be a filtered simplicial complex. Then the *isomorphism class of filtered simplicial complexes* $([X], F)$ is the equivalence class such that the ordering created by the filtration function F on X is preserved by all simplicial maps between members of the isomorphism class $[X]$.

It is important to notice that if two filtrations $F\colon X \to \mathbb{R}$ and $G\colon X \to \mathbb{R}$ on a simplicial complex X induce distinct orderings on the simplices of X, the isomorphism classes $([X], F)$ and $([X], G)$ are also distinct. However, if the two filtrations produce the same total ordering on X, then the filtered simplicial complexes (X, F) and (X, G) will be members of the same isomorphism class $([X], F) = ([X], G)$.

Definition 2.20. Define the *set of all isomorphism classes of finite filtered simplicial complexes* to be

$$(X^*, F) = \{([X], F) \mid [X] \in X^* \text{ and } F \text{ a filtration on X}\}.$$

The set of all isomorphism classes of finite filtered simplicial complexes represents the most generalized similarities between filtered, simplicial complexes as it reduces simplicial complexes to common isomorphism classes and filtrations to isomorphic total orderings.

We may also use the language of category theory to describe finite filtered simplicial complexes.

Definition 2.21. The category XF is the *category of finite filtered simplicial complexes*. Its objects are the finite filtered simplicial complexes (X, F) and its morphisms are simplicial maps $\phi\colon X \to Y$ such that, given two finite filtered simplicial complexes (X, F) and (Y, G), $F(\sigma) \geq G(\phi(\sigma))$ for all $\sigma \in X$.

Checking that this does constitute a category, consider the morphisms ϕ between (X, F) and (Y, G) and ψ between (Y, G) and (Z, H). Notice $F(\sigma) \geq G(\phi(\sigma))$ for all $\sigma \in X$ and $G(\tau) \geq H(\psi(\tau))$ for all $\tau \in Y$. Then as

$$F(\sigma) \geq G(\phi(\sigma)) \geq H(\psi(\phi(\sigma)))$$

for all $\sigma \in X$, we have a composition rule for morphisms.

Associativity follows as the morphisms are simplicial maps. Given three compatible morphisms $(X, F) \xrightarrow{\phi} (Y, G)$, $(Y, G) \xrightarrow{\psi} (Z, H)$, and $(Z, H) \xrightarrow{\theta} (W, E)$ we have

$$\theta \circ (\psi \circ \phi)(\sigma) = \theta(\psi(\phi(\sigma))) = (\theta \circ \psi) \circ \phi(\sigma)$$

for all $\sigma \in X$. Finally, given (X, F), the identity morphism $\mathrm{id}_{(X,F)}$ is simply the inclusion simplicial map from X to itself. Thus, XF is in fact a category.

2.3 Homology

Homology is a mathematical framework for discussing, unambiguously, how a topological space is connected. The most intuitive way to understand homology is by connected components and higher dimensional holes in a space. Consider the boundary of the standard 2-simplex in Figure 2. We will find that the only nontrivial homology groups are H_1 and H_0 and in fact each give one copy of the field used for this computation. Topologically speaking, this translates to our complex consisting of a single 1-dimensional hole (the loop formed by the 1-simplices) and a singular connected component.

Circumventing the philosophical discussion of the metaphysical characterization of holes [13], homology groups take an indirect approach at the discovery and classification of a space's holes by focusing instead on what surrounds them. This is done using sequences called *chain complexes.*

Definition 2.22. A *chain complex* $\mathcal{C}$ is a sequence of abelian groups, called *chain groups*,

$$\cdots \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} \cdots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} \cdots$$

connected by homomorphisms $\partial_k \colon C_k \to C_{k-1}$ such that $\partial_{k-1}\partial_k = 0$ for all k. Note that $C_{-1} = \emptyset$ and the individual elements of each *chain group* C_p are called *p-chains.*

In order to adapt these chain complexes for the abstract simplicial complexes discussed in Section 2.1 we need the following.

Definition 2.23. An *orientation* of an n-simplex is an ordering of its vertices $[v_0, v_1, ..., v_n]$. Any two orientations on the same simplex are equivalent if there is an even permutation that changes one to the other.

Defining orientations on simplices allows us to construct explicit boundary maps with the purpose of ensuring the composition criterion of Definition 2.22.

Definition 2.24. Let X be a simplicial complex and σ be an oriented n-simplex of X given by $[v_0, v_1, ..., v_n]$. Then the *boundary map* ∂_n is defined as follows:

$$\partial_n(\sigma) = \partial_n([v_0, v_1, ..., v_n]) = \sum_{i=0}^{n} (-1)^i \sigma|_{[v_0,...,\hat{v_i},...,v_n]}$$

where $\sigma|_{[v_0,...,\hat{v_i},...,v_n]}$ is the simplex created by removing the i^{th} vertex from the simplex σ.

Now given an abstract simplicial complex (X, V), we choose $C_p(X)$ to be the free abelian group of oriented p-simplices and the p-chains as formal linear combinations of p-simplices. We claim that the boundary map in definition 2.24 acts as the desired homomorphism for our chain complex. We must now verify the boundary map is 0 for double composition.

Proposition 2.25. *The composition $\partial_{n-1} \circ \partial_n$ is always zero.*

Proof. Let σ be an oriented n-simplex in the simplicial complex X. We write $\sigma = [v_0, v_1, ..., v_n]$ and, by definition, $\partial_n(\sigma) = \sum_i (-1)^i \sigma|_{[v_0,...,\hat{v_i},...,v_n]}$. Now applying ∂_{n-1} we get

$$\begin{aligned}
\partial_{n-1}(\partial_n(\sigma)) &= \sum_j (-1)^j \left(\sum_i (-1)^i \sigma|_{[v_0,...,\hat{v_i},...,v_n]} \right) \Big|_{[v_0,...,\hat{v_j},...,v_n]} \\
&= \sum_j \sum_i (-1)^j (-1)^i \left(\sigma|_{[v_0,...,\hat{v_i},...,v_n]} \right) |_{[v_0,...,\hat{v_j},...,v_n]} \\
&= \sum_{j<i} (-1)^i (-1)^j \sigma|_{[v_0,...,\hat{v_j},...,\hat{v_i},...,v_n]} \\
&\quad + \sum_{i<j} (-1)^i (-1)^{j-1} \sigma|_{[v_0,...,\hat{v_i},...,\hat{v_j},...,v_n]}
\end{aligned}$$

where the second term in the sum can be expressed as the negative of the first when indexing over all $j < i$ instead of $i < j$. This demonstrates that the double composition of ∂_n and ∂_{n-1} is always zero and thus the boundary maps $\{\partial_n\}_{n\in\mathbb{N}}$ define the chain complex $\mathcal{C}(X) = \{C_n(X)\}_{n\in\mathbb{N}}$ for the simplicial set (X, V). □

The following terminology is commonly used when discussing the boundary maps: $Z_p = \ker(\partial_p)$ and $B_p = \text{im}\,(\partial_{p+1})$. We must also distinguish two special types of chains in a chain complex. The first being *p-cycles* that are p-chains with

empty boundaries, $\partial_p c = 0$, that reside in Z_p. The second, *p-boundaries*, reside in B_p and are defined by being the boundary of a $(p+1)$-chain, $c = \partial_{p+1} c'$ for some $c' \in C_{p+1}$.

As we have shown, the image of a p-boundary under the boundary map is always zero. Hence, the group of p-boundaries is a subgroup of the group of p-cycles. Thinking intuitively, a p-hole must be confined by some p-cycle. However, not all p-cycles surround a hole. The boundary of a $(p+1)$-simplex is a p-cycle by Proposition 2.25, yet by definition it could not confine a hole as its interior is filled by our $(p+1)$-simplex. In order to find the holes in our complex, we must restrict our p-cycles to those that do not serve as the p-boundary for some other simplex. Hence, we take the quotient group of the p-cycles modulo p-boundaries. This leads us directly to the *homology groups*.

Definition 2.26. The p^{th} *homology group* of a chain complex is defined as

$$H_p = \frac{Z_p}{B_p} = \frac{\ker(\partial_p)}{\text{im}\,(\partial_{p+1})}.$$

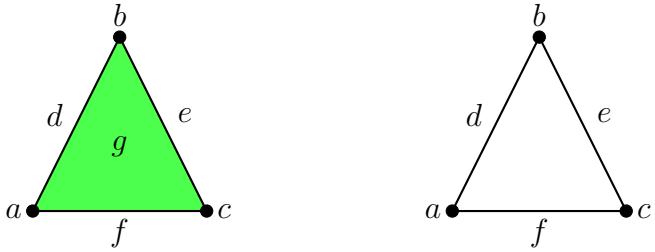


Figure 10: The standard 2-simplex Δ^2 (left) and its boundary $\partial\Delta^2$ (right). The simplices are given alphabetical signifiers to more precisely refer back to them.

Consider two closely related examples: the standard 2-simplex and its boundary, see Figure 10. We will first use the chain complex for $\partial\Delta^2$ to compute the homology groups. We represent the free abelian groups of oriented simplices by their respective set of generators. The only non-trivial chain groups are $C_0(\partial\Delta^2) = \{a, b, c\}$ and $C_1(\partial\Delta^2) = \{d, e, f\}$. Notice the only 1-cycles are those in the subgroup $Z_1 = \{d - e + f\}$ and there are no 1-chains that are also the boundary of a 2-chain. For $C_0(\partial\Delta^2)$, we can make the following identifications:

$$\begin{aligned} b &= b - a + a = -\partial_1(d) + a \\ c &= c - a + a = -\partial_1(f) + a \\ \partial(e) = b - c &= (a - c) + (b - a) = \partial_1(f) - \partial_1(d). \end{aligned}$$

Applying these to $C_0(\partial\Delta^2)$, we find that

$$C_0(\partial\Delta^2) = \{a, -\partial_1(d) + a, -\partial_1(f) + a\} = \{a, \partial_1(d), \partial_1(f)\}$$

and

$$\begin{aligned} Z_0 &= \{a, \partial_1(d), \partial_1(f)\} \\ B_0 &= \{\partial_1(d), \partial_0(f)\}. \end{aligned}$$

Using Definition 2.26 to compute the 0^{th} and 1^{st} homology groups of our complex, we find of the quotient groups each have only one generator. This translates to there being one 0-homology feature (the connected component) and one 1-homology feature (the loop formed by the three 1-simplices). This is in agreement with both the discussion at the beginning of this section and our intuition for the characterization of $\partial\Delta^2$'s holes.

Now consider what happens when we decide to include the 2-simplex in our complex. We expect that the 1-dimensional hole we found in $\partial\Delta^2$ will vanish leaving only a single connected component. Notice that $C_2(\Delta^2)$ is no longer trivial and 1-cycles generated by $d - e + f$ are now also 1-boundaries of the 2-chains generated by g. This tells us that $Z_1 = \{d - e + f\} = B_1$ and so H_1 is now trivial. As g has a non-zero image under ∂_2, H_2 will also be trivial. Using the same identifications and argument as with $\partial\Delta^2$, H_0 has exactly one generator.

2.4 Persistent Homology

The homology groups that we have found give us intuition about the important topological structure of a given abstract simplicial complex. It is, however, simply a snapshot in time and lacks information on important topological features that are created and destroyed during the construction of our specific complex. Let us again consider the 2-simplex in Figure 10. Having computed the various homology groups, $H_p(\Delta^2)$, of this structure, we know that only $p = 0$ is a non-trivial group. Yet if we are to build up this simplicial complex Δ^2 one simplex at a time we will eventually need to pass through the simplicial subcomplex $\partial\Delta^2$, which we know has a non-trivial $H_1(\partial\Delta^2)$ group.

This leads us to ask: how can we account for the homological features that arise and vanish throughout the construction of a given complex? The answer is persistent homology.

Consider a finite filtered simplicial complex (X, F). Letting there be m simplices in X, the filtration function $F\colon X \to \mathbb{R}$ defines a sequence of $n + 1 \leq m + 1$ subcomplexes of X

$$\emptyset = X_0 \subseteq X_1 \subseteq \cdots \subseteq X_n = X$$

such that the maximum value under F of all simplices in X_i is a_i and $a_0 < a_1 < \cdots < a_n$. For every $i \leq j$, we have an inclusion map from X_i to X_j and hence an induced homomorphism $f_p^{i,j}: H_p(X_i) \to H_p(X_j)$ for each dimension p. From our sequence of subcomplexes, we find an induced sequence of homology groups connected by homomorphisms

$$0 = H_p(X_0) \to H_p(X_1) \to \cdots \to H_p(X_n) = H_p(X)$$

for each dimension p. This induced sequence of homology groups is the key to understanding the evolution of topological features as our original finite filtered simplicial complex is constructed.

Definition 2.27. Given a finite filtered simplicial complex (X, F), the p^{th} *persistent homology groups* are the images of the homomorphisms, $f_p^{i,j}: H_p(X_i) \to H_p(X_j)$ for $0 \leq i \leq j \leq n$ induced by the inclusion of $X_i \hookrightarrow X_j$.

We define the homomorphism $f_p^{i,i}$ as the identity on $H_p(X_i)$ for all i and p. Letting c be a homology class in $H_p(X_i)$, c is *born* at X_i if $c \notin \operatorname{im}(f_p^{i-1,i})$. If the homology class c is born at X_i, it is said to *die entering* X_j if it is absorbed by another homology class as we go from X_{j-1} to X_j. When two classes merge, the class that absorbs or kills the other is determined by which class was born first, hence the name: the *Elder Rule* [8].

When looking at a specific homology class c that is born at X_i and dies entering X_j, its birth and death information are combined in the *persistence pair* (a_i, a_j). We call the filtration value a_i, corresponding to X_i, the *birth time* of c and the filtration value a_j is recorded as our class' *death time*. The *persistence* of a class c is defined as $a_j - a_i$. In the case when c never dies entering a subcomplex X_j, it is given an infinite persistence and is called a *persistent feature* of (X, F).

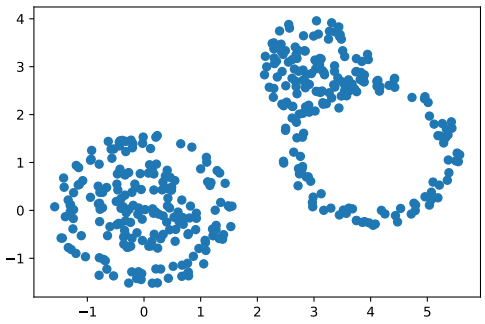


Figure 11: Example point cloud embedded in $\mathbb{R}^2$.

There are two ways to present the persistent homology information for a simplicial complex. One is the *persistent diagram* that plots each homology feature, persistent or otherwise, by the pair of its birth and death times. The other is the *barcode diagram*, which encodes each homological feature as a line segment, beginning at its birth time and terminating at its death time. Each encode the same topological information, but present them slightly differently.

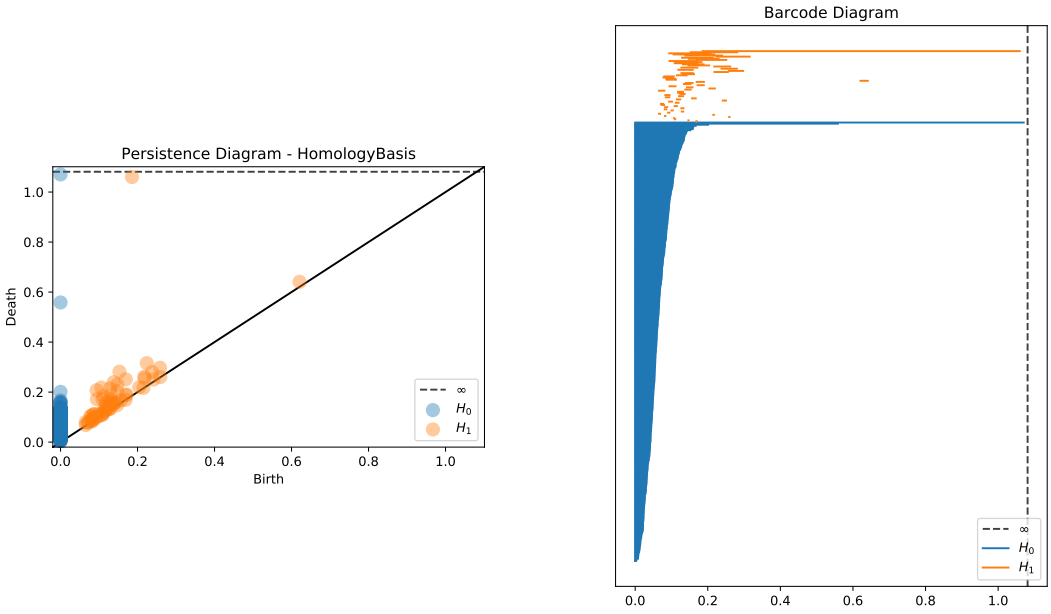


Figure 12: The persistence (left) and barcode (right) diagrams for the Čech progression of Figure 11 restricted to simplices of dimension 0, 1, and 2.

For an example of each, consider varying the parameter of the Čech complex from 0 to ∞ for the data set shown in Figure 11. As the data set is embedded in $\mathbb{R}^2$, we have restricted the complex to simplices of dimension 0, 1, and 2. The persistence and barcode diagrams are produced in Figure 12. In the persistence diagram, all points that fall close to the line $x = y$ are most likely noise that is unimportant to the fundamental structure of our complex. The analog of this in the barcode diagram are bars that have a very short length. There are, however, three features that we consider interesting. First is the persistent 0-homology feature that will always be present. This is denoted by the 0-homology point in the up left corner of our persistence diagram and by the longest 0-homology bar in our barcode diagram. The second is the 0-homology feature that has a birth

time of 0 and a death time of roughly 0.55. This is a much larger persistence than the majority of the other 0-homology features and so we may intuit that there are two main connected components that make up our simplicial complex. Lastly, we have a 1-homology feature that persists much longer than the other features of this dimension. It is represented by the 1-homology point near the top of our persistence diagram and by the longest 1-homology bar in our barcode diagram. It is not a persistent feature which tells us that it is not present in the $\check{\mathrm{C}}_\infty$ complex, but appears in a large number of the subcomplexes we built along the way. Hence, we may guess that there is a prominent loop in our data set. Looking back to our point cloud in Figure 11, we are able to identify the key features that our analysis of the persistence and barcode diagrams picked out.

2.5 Totally Filtered Chain Complexes

We now introduce our main algebraic structure, the *totally filtered chain complex*. As its name suggests, the totally filtered chain complex combines the chain complex from Section 2.3 with the the idea of using filtrations to assess how the homology groups evolve over the construction of the various subcomplexes from Section 2.4. Recall, given a field k and function $f\colon X \to k$, the set of all elements of X with non-zero image under f is called the support of f.

Definition 2.28. Given a set X and field k, the *free k-vector space $k\{X\}$ based on X* is the set of functions $v\colon X \to k$ with finite support.

Definition 2.29. A *totally filtered chain complex* based on a totally ordered set $[n]$ is the pair $C = (\partial, \deg)$ of a k-linear map $\partial\colon k\{[n]\} \to k\{[n]\}$ and a function $\deg\colon [n] \to \mathbb{Z}$. Writing $F_iC_p = k\{[i] \cap \deg^{-1}(p)\}$, we require:

1. $\partial \circ \partial = 0$.
2. For all $0 \leq i \leq n$ and $p \in \mathbb{Z}$, the k-linear map ∂ restricts to a k-linear map $\partial_{i,p}\colon F_iC_p \to F_{i-1}C_{p-1}$.

We write $C_p = k\{[n] \cap \deg^{-1}(p)\}$ and $\partial_p\colon C_p \to C_{p-1}$ for the restriction of ∂. As with other chain complexes, $Z_p = \ker(\partial_p)$ and $B_p = \operatorname{im}(\partial_{p+1})$.

Given $i \in \mathbb{N}$ and a totally filtered chain complex $C = (\partial, \deg)$, we denote the totally filtered chain complex based on the restriction to $[i] \cap [n]$ with $F_iC = (F_i\partial, F_i \deg)$. In F_iC, $F_i\partial$ is given by the restriction of ∂ to the k-linear subspace $k[i] \cap [n]$ and $F_i \deg$ is the restriction of deg to the subspace $[i] \cap [n]$ of $[n]$.

We now introduce an injective function between the set of all isomorphism classes of finite filtered simplicial complexes and the set of all totally filtered chain complexes. It is believed that this relationship may be more aptly described as a

functor between the category of filtered simplicial complexes (see Definition 2.21) and the category of totally filtered chain complexes. However, this requires a precise description of the latter category which has not been created up to this point.

Proposition 2.30. *Let* TFCC *be the set of totally filtered chain complexes. Choosing a field k, there exists a injective function*

$$\Gamma_k \colon (X^*, F) \to \text{TFCC}$$

that uniquely maps each isomorphism class of finite filtered simplicial complex to a totally filtered chain complex.

Proof. We first propose a schema for the function Γ_k and then prove that this construction is in fact injective.

Let $([X], F)$ be a isomorphism class of finite filtered simplicial complexes, with representative (X, F). Then $F \colon X \to \mathbb{R}$ defines a total ordering on X such that $F(\sigma) \leq F(\tau)$ implies $\sigma \subseteq \tau$ for all $\tau, \sigma \in X$. As X is finite, we know this is a finite total ordering on $n+1$ simplices for some non-negative integer n. The total order induces an isomorphism $I \colon X \xrightarrow{\cong} [n]$. Note that this isomorphism is unique up to reordering. From I we derive an appropriate $\deg \colon [n] \to \mathbb{Z}$ function, namely one that preserves the degrees of all simplices in X under I. Now to find the k-linear map ∂. As we have already chosen the field k that we wish to work in, we know we must work in the free k-vector space $k\{[n]\}$. This k-vector space is spanned by basis functions of the form $\mathbf{f}_\sigma$ such that

$$\mathbf{f}_\sigma(\tau) = \begin{cases} 1 & \text{for } \tau = \sigma \\ 0 & \text{else} \end{cases}$$

Any function defined on the free k-vector space is uniquely described by how it acts on the basis elements. Given a p-simplex $x \in (X, F)$ with orientation $x = [v_{x_0}, v_{x_1}, \ldots, v_{x_{p-1}}]$, any of the boundary $(p-1)$-simplices' orientations may be denoted by $dx_i = [v_{x_0}, \ldots, \hat{v}_{x_i}, \ldots, v_{x_{p-1}}]$ where v_{x_i} is removed. Let $\sigma = I(x)$, that is σ is the representative of the simplex x under I. We define $\partial \colon k\{[n]\} \to k\{[n]\}$ by

$$\partial(\mathbf{f}_\sigma) = f_{\partial(\sigma)} \colon [n] \to k$$

where

$$\mathbf{f}_{\partial(\sigma)}(\tau) = \begin{cases} 1 & \text{if } \tau = I(dx_i) \text{ for some even } i \\ -1 & \text{if } \tau = I(dx_i) \text{ for some odd } i \\ 0 & \text{else} \end{cases}$$

We now check that our proposed $(\partial, \deg)$ is in fact a totally filtered chain complex. Consider the basis element in $k\{[n]\}$, $\mathbf{f}_\sigma$, for some $\sigma = I(x)$ and $x \in (X, F)$. Then

$$\partial(\mathbf{f}_\sigma) = f_{\partial(\sigma)} = \sum_i (-1)^i f_{I(dx_i)}.$$

Both $dI^{-1}(I(dx_i))_j$ and $dI^{-1}(I(dx_j))_i$ pick out the same oriented simplex in (X, F), however it is combinatorially important to track whether $i < j$ or $j < i$. We use the notation scheme that $dx_{i,j}$ refers to the stated $(p-2)$-simplex when $i < j$. Now when taking the double boundary of $\mathbf{f}_\sigma$ we get

$$\begin{aligned}
\partial \circ \partial(\mathbf{f}_\sigma) = \partial(\mathbf{f}_{\partial(\sigma)}) &= \partial(\sum_i (-1)^i \mathbf{f}_{I(dx_i)}) \\
&= \sum_i (-1)^i \partial(\mathbf{f}_{I(dx_i)}) \\
&= \sum_i (-1)^i (\sum_{i \neq j} (-1)^j \mathbf{f}_{I(dI^{-1}(I(dx_i))_j)}) \\
&= \sum_{j<i} (-1)^{i-1}(-1)^j \mathbf{f}_{I(dx_{j,i})} + \sum_{i<j} (-1)^i (-1)^j \mathbf{f}_{I(dx_{i,j})}
\end{aligned}$$

By the same argument as in the proof of Proposition 2.25 the two terms sum to zero.

Let $0 \leq i \leq n$ and $p \in \mathbb{Z}$. We hope to show that ∂ restricts to $\partial_{i,p} \colon F_i C_p \to F_{i-1} C_{p-1}$. As the isomorphism I is induced by the total ordering applied by the filtration F on X, we know that if $\{i\}$ is the representative of some simplex $x \in (X, F)$ then a face of x must be represented by $\{j\}$ such that $j < i$. Also, by definition, deg preserves the degree structure from (X, F) and ∂ is defined using dx_i's which are of one lower degree than x. As ∂ restricts individually for both i and p, it also restricts for them together. Thus we have constructed a schema to send $([X], F)$ into a totally filtered chain complex based on $[n]$ in the field k.

The injectivity of Γ_k follows from the fact that the isomorphism $I \colon X \to [n]$ is unique up to reordering. Let (X, F) and (Y, G) be two finite filtered simplicial complexes such that $\Gamma_k(([X], F)) = \Gamma_k(([Y], G))$. We have an isomorphism between X and Y, as both have isomorphisms, $I_X \colon X \to [n]$ and $I_Y \colon Y \to [n]$ respectively, between them and the totally ordered set $[n]$. As I_X and I_Y are induced by the filtration functions F and G, the isomorphism extends to the filtration functions as well. As the degree and boundary relations that are encoded in the totally filtered chain complex are the same for both $([X], F)$ and $([Y], G)$, we may construct an isomorphic simplicial map between the simplicial complexes X and Y. Thus $([X], F) = ([Y], G)$, demonstrating the injectivity of Γ_k. □

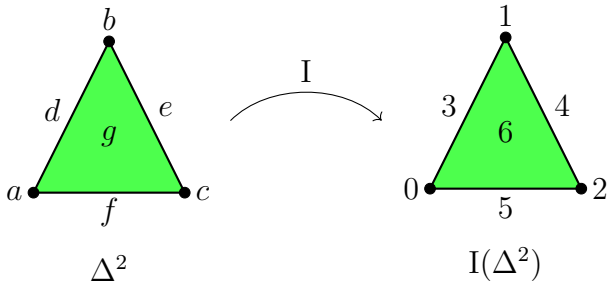


Figure 13: The standard 2-simplex Δ^2 along with the images of its simplices under the isomorphism $I\colon \Delta^2 \to [6]$.

Example 2.31. Let us use Proposition 2.30 to construct a totally filtered chain complex from the standard 2-simplex, see Figure 13. We apply a general filtration to Δ^2 such that $F(a) < F(b) < \cdots < F(f) < F(g)$.

First we choose the field we wish to be working in. For simplicity we select $\mathbb{Z}_p$ for some prime p. Next we choose the totally ordered set to be $[6]$ as we have 7 simplices in our complex. The isomorphism $I\colon \Delta^2 \to [6]$ is constructed as in Figure 13. We now define the degree and boundary functions for our totally filtered chain complex on the basis elements of $\mathbb{Z}_p\{[6]\}$ in the table below.

σ	$I(\sigma)$	$\deg(I(\sigma))$	$f_{I(\sigma)}$	$\partial(f_{I(\sigma)})$
a	0	0	$\mathbf{f}_0$	0
b	1	0	$\mathbf{f}_1$	0
c	2	0	$\mathbf{f}_2$	0
d	3	1	$\mathbf{f}_3$	$\mathbf{f}_0 - \mathbf{f}_1$
e	4	1	$\mathbf{f}_4$	$\mathbf{f}_1 - \mathbf{f}_2$
f	5	1	$\mathbf{f}_5$	$\mathbf{f}_0 - \mathbf{f}_2$
g	6	2	$\mathbf{f}_6$	$\mathbf{f}_3 - \mathbf{f}_4 + \mathbf{f}_5$

Here $\mathbf{f}_i$ is the basis element of $\mathbb{Z}_p\{[6]\}$ for $i \in [6]$ and $0\colon [6] \to \mathbb{Z}_p$ is the function such that $0(x) = 0$ for all $x \in [6]$. As we used our schema from Proposition 2.30 to construct them, the functions deg and ∂ define a totally filtered chain complex that stores the same combinatorial information as Δ^2.

3 Homology Bases

The main purpose of this thesis is to devise a way to efficiently compute homology groups. In Section 2.5 we introduced our main algebraic structure, the totally filtered chain complex, and showed that for each isomorphism class of finite filtered simplicial complexes there exist unique totally filtered chain complex that encodes the same combinatorial information. Converting a finite filtered simplicial complex in this way allows us to disregard all information that does not directly contribute to the complex's homology groups. Similar to how these homology groups are not always immediately apparent given an abstract simplicial complex, we must suss out the desired homology from our totally filtered chain complexes. We now introduce the main mechanism used for this task: the *homology basis*.

3.1 The Homology Basis

Recall a k-linear map $f: V \to V$ is a *projection* if $f \circ f = f$.

Definition 3.1. Let P be a subset of $[n]$, $C = (\partial, \deg)$ be a totally filtered chain complex based on $[n]$, and $f: k\{[n]\} \to k\{[n]\}$ be a k-linear projection such that $\operatorname{im}(f) = k\{P\}$. Then (P, f) is a *homology basis* of C if all the following hold:

1. f preserves degree,
2. $f(Z_p) = k\{P \cap \deg^{-1}(p)\}$, and
3. $\ker(f) \cap Z_p = \operatorname{im}(\partial_{p+1})$.

The following theorem demonstrates how the homology basis of a totally filtered chain complex successfully extracts the essential homological information we seek.

Theorem 3.2. *A homology basis (P, f) for C induces an isomorphism with each homology group, $H_*(C) \to k\{P\}$.*

Proof. Given a totally filtered chain complex C, we have the short exact sequence with H_p the p^{th} homology group. Let $f|_{Z_p}$ be the function f restricted to Z_p. Then $\ker(f|_{Z_p}) = \operatorname{im}(\partial_{p+1}) = B_p$ and $Z_p/\ker(f|_{Z_p}) \cong k\{P \cap \deg^{-1}(p)\}$ gives us that H_p is isomorphic to $f(Z_p)$ as seen below.

$$\begin{array}{ccccccccc} 0 & \longrightarrow & B_p & \longrightarrow & Z_p & \longrightarrow & H_p = Z_p/B_p & \longrightarrow & 0 \\ & & \downarrow{=} & & \downarrow{=} & & \downarrow{\cong} & & \\ 0 & \longrightarrow & ker(f|_{Z_p}) & \longrightarrow & Z_p & \longrightarrow & k\{P \cap deg^{-1}(p)\} & \longrightarrow & 0 \end{array}$$

Thus (P, f) contains all homology information. □

Corollary 3.3. *The underlying set P of a homology basis (P, f) is unique.*

Proof. Let Q and P be two underlying sets for a homology basis of a totally filtered chain complex C. Then by Theroem 3.2

$$k\{Q\} \cong H_*(C) \cong k\{P\}$$

where

$$k\{Q\} = \{g \colon Q \to k \mid \text{support}(g) \text{ is finite}\}$$

and

$$k\{P\} = \{h \colon P \to k \mid \text{support}(h) \text{ is finite}\}.$$

Thus the underlying sets Q and P must be isomorphic. □

While the underlying set P of a homology basis will always be unique, the same does not hold for the k-linear projection.

Proposition 3.4. *The k-linear projection $f \colon k\{[n]\} \to k\{[n]\}$ of a homology basis is not unique.*

Proof. Let C be based on $[n]$ and (P, f) be the homology basis of C. Let k be the fixed field. Choose an element $0 \neq \alpha \in k$. Now consider the projection αf that is, given functions $\psi \in k\{[n]\}$ and $\psi' = f(\psi) \in k\{P\}$, the map $\psi \mapsto \alpha\psi'$ with $\alpha\psi'$ the map $x \mapsto \alpha\psi'(x)$. Now we check if $(P, \alpha f)$ is also a homology basis.

Multiplying a function in $k\{P\}$ by an element of k does not change the set that the free k-vector space is based on. Given $\psi' \in k\{P\}$, the function $\alpha\psi'$ then stays in $k\{P\}$. Hence, if $\text{im}\,(f) = k\{P\}$, then $\text{im}\,(\alpha f) = k\{P\}$. If f is degree preserving then αf must also be degree preserving as each image is a linear combination of the same basis elements. The image of the kernel of ∂_p is unaffected by the choice of f. If criterion (2), in Definition 3.1, holds for f it holds for αf. By construction the same basis elements must be taken to zero by both f and αf thus the intersection of the kernel and Z_p is unchanged and $\ker(\alpha f) \cap Z_p = \text{im}\,(\partial_{p+1})$. Thus, if (P, f) is a homology basis so is $(P, \alpha f)$. □

Example 3.5. Consider the selection of subcomplexes of the standard 2-simplex shown in Figure 14. We give the implicit index filtration to our complexes and use Proposition 2.30 to generate a totally filtered chain complex for the standard 2-simplex such that each of our subcomplexes is a restriction $F_i\Delta^2$. We will denote the homology bases from left to right as (P_2, f_2) up to (P_5, f_5). As $F_2\Delta^2$ is only three disjoint simplices the k-linear projection for the homology basis is the identity. As the projection is the identity, all the criteria for a homology basis are fulfilled and $(P_2, f_2) = (\{0, 1, 2\}, \text{id})$. Moving to $F_3\Delta^2$ we find that the newly added 1-simplex has forced the x and y vertices into the same connected component. The

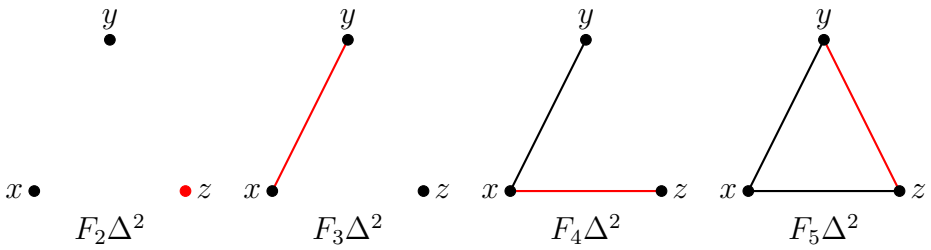


Figure 14: Selection of subcomplexes of the standard 2-simplex with the last added simplex highlighted in red.

x and y representatives $0, 1 \in [n]$ can no longer generate distinct 0-homology features. Using the Elder Rule, we know that the homology feature generated by the basis element 1 dies while entering $F_3\Delta^2$. We then adjust our projection to represent this and find our new homology basis is $(\{0, 2\}, f_3)$ where f_3 is the previous projection f_2 now sending $\mathbf{f}_1 = \max(\text{support}(\partial(\mathbf{f}_3)))$ to $\mathbf{f}_0$ as well as $f_3(\mathbf{f}_3) = 0$. We repeat the same approach when entering $F_4\Delta^2$ to find $(P_4, f_4) = (\{0\}, f_4)$ where $f_4(\mathbf{f}_1) = f_4(\mathbf{f}_2) = \mathbf{f}_0$ and all other elements are taken to 0. However, when looking at $F_5\Delta^2$ we find a new 1-homology feature is born leading us to expand our basis from (P_4, f_4). This is because when looking at $\mathbf{f}_4(\partial(\mathbf{f}_5))$ we find it is zero in P_4. Thus, we have connected simplices that are already in the same homology class (forming a new 1-cycle) and creating a new homology class. Hence, we find the homology basis $(\{0, 5\}, f_5)$ with f_5 being the unique extension of f_4 to $F_5\Delta^2$ with $f_5(\mathbf{f}_5) = \mathbf{f}_5$.

In the last example, when transitioning from the $F_4\Delta^2$ totally filtered chain complex to $F_5\Delta^2$, we find that a new homology feature has been created. In a similar vein to the discussion of evolving homological features back in Section 2.4, we say this feature was *born* at $F_5\Delta^2$ and define its *birth time* in our totally filtered chain complex as follows.

Definition 3.6. Given a totally filtered chain complex $C = (\partial, \deg)$ based on $[n]$, the *birth time* of a homology class $c \in H_pC$ is the minimal element i of $[n]$ with the property that c is in the image of the homomorphism $H_p(F_iC) \to H_p(C)$ induced by the inclusion $F_iC \subseteq C$.

Back in Section 2.4, we defined birth time as the filtration value associated with the subcomplex that the homology class is born in. This seems to conflict with Definition 3.6. However, if our totally filtered chain complex C represents a finite filtered simplicial complex (X, F), we can utilize the isomporhism $I\colon X \to [n]$ to jump between the two definitions. Specifically, given a birth time i for a class $c \in H_p(C)$, we find the filtration value of this birth time in (X, F) is $F(I^{-1}(i))$.

Returning to Example 3.5, we see that each transition from $F_i\Delta^2$ to $F_{i+1}\Delta^2$ produced a change in the underlying set P_i that was either the addition of a new element or the removal of an existing one. These transitions are called *homology basis extensions* and *homology basis contractions*, respectively.

Definition 3.7. Let (Q, g) and (P, f) be homology bases of $F_{i-1}C$ and F_iC respectively, such that Q is a proper subset of P. If q is the homomorphism $q(j) = j$ induced by the inclusion of Q in P and the diagram

$$\begin{array}{ccc} H_*(F_{i-1}C) & \xrightarrow{\quad h \quad} & H_*(F_iC) \\ \downarrow{\scriptstyle \overline{g}} & & \downarrow{\scriptstyle \overline{f}} \\ k\{Q\} & \xrightarrow{\quad q \quad} & k\{P\} \end{array}$$

commutes, where h is the homomorphism induced by the inclusion $F_{i-1}C \subset F_iC$, $\overline{g}$ and $\overline{f}$ are isomorphisms as in Theorem 3.2, then q is a *homology basis extension*.

Definition 3.8. Let (Q, g) and (P, f) be homology bases of $F_{i-1}C$ and F_iC respectively such that P is a proper subset of Q. If q is the homomorphism

$$q(j) = \begin{cases} j & \text{if } j \in P \\ 0 & \text{if } j \notin P \end{cases}$$

and the diagram

$$\begin{array}{ccc} H_*(F_{i-1}C) & \xrightarrow{\quad h \quad} & H_*(F_iC) \\ \downarrow{\scriptstyle \overline{g}} & & \downarrow{\scriptstyle \overline{f}} \\ k\{Q\} & \xrightarrow{\quad q \quad} & k\{P\} \end{array}$$

commutes, where h is the homomorphism induced by $F_iC \subset F_{i-1}C$, $\overline{g}$ and $\overline{f}$ are isomorphisms as in Theorem 3.2, then q is a *homology basis contraction*.

We now introduce a classification for homology bases that are the most desirable for the purpose of extracting homological information. That is the *minimal* homology basis.

Definition 3.9. A homology basis (P, f) of a totally filtered chain complex $C = (\partial, \deg)$ based on $[n]$ is *minimal* if for every homology class $c \in H_p(C)$, the birth time i of c can be described as follows: under the isomorphism

$$H_p(C) \xrightarrow{\overline{f}_p} k\{P \cap \deg^{-1}(p)\},$$

the homology class c is taken to an element $\overline{f}_p(c)$ with the property that the maximal element in the support of $\overline{f}_p(c)$ is i. In particular $i \in P$.

Working with minimal homology bases is desirable because it ensures that the image of every homology class c under $\overline{f}$ requires no additional information from homology classes with birth time greater than itself.

Proposition 3.10. *The inclusion $H_*(F_{i-1}C) \subset H_*(F_iC)$ induces a homology basis extension when $F_{i-1}C$ has a minimal homology basis (Q, g).*

Proof. Let $F_{i-1}C$ and F_iC be subsequent totally filtered chain complexes such that $H_*(F_{i-1}C) \subset H_*(F_iC)$. We know that $F_{i-1}C$ has a minimal homology basis (Q, g) and the inclusion of the simplex $\{i\}$ corresponds to the creation of a new homology feature.

First construct a homology basis for F_iC. Choose $P = Q \cup \{i\}$ and let f be the unique extension of g to F_iC such that $f(\{i\}) = \{i\}$. Clearly f preserves degree as g was degree preserving. To see $f \circ f = f$ with $\operatorname{im}(f) = k\{P\}$, notice

$$\operatorname{im}(g) \cup \{i\} = k\{Q \cup \{i\}\} = k\{P\}.$$

We see

$$f(Z_p) = k\{P \cap \deg^{-1}(p)\}$$

for all p as $f(Z_p)$ is unchanged for p up to $p = \deg(i) - 1$ and

$$f(Z_{\deg(i)}) = k\{P \cap \deg^{-1}(\deg(i))\}$$

holds given our change (Q, g) to (P, f). Since $\partial\{i\} \in \ker(g) \cap F_{i-1}Z_{\deg(i)-1}$ and $\ker(g) \cap F_{i-1}Z_{\deg(i)-1} = \operatorname{im}(F_{i-1}\partial_{deg(i)})$, we have $\operatorname{im}(F_{i-1}\partial_p) = \operatorname{im}(\partial_p)$ for all p. As $\ker(g) = \ker(f)$, we see that $\ker(f) \cap Z_p = \operatorname{im}(\partial_{p+1})$ and that (P, f) is a homology basis of F_iC.

Now define the homomorphism $q\colon k\{Q\} \to k\{P\}$ induced by the inclusion $Q \subset P$ such that $q(j) = j$. As (Q, g) is minimal and the birth time of $\partial\{i\}$ is i, (P, f) is also minimal. Hence, the diagram in Definition 3.7 commutes when h is the homomorphism induced by the inclusion $F_{i-1}C \subset F_iC$ and both $\overline{g}$ and $\overline{f}$ are isomorphisms as in Theorem 3.2. Thus, q is a homology basis extension induced by the inclusion $H_*(F_{i-1}C) \subset H_*(F_iC)$. □

A similar result is obtained when the proper inclusion is reversed. This time inducing a homology basis contraction.

Proposition 3.11. *The inclusion $H_*(F_iC) \subset H_*(F_{i-1}C)$ induces a homology basis contraction when $F_{i-1}C$ has a minimal homology basis (Q, g).*

Proof. Let $F_{i-1}C$ and F_iC be subsequent totally filtered chain complexes such that $H_*(F_iC) \subset H_*(F_{i-1}C)$. The inclusion of the simplex $\{i\}$ corresponds to the destruction of a homology feature in $F_{i-1}C$.

We now construct a homology basis for F_iC from (Q, g). Let $v = g(\partial\{i\})$ and l be the maximum element in the support of v with respect to the homology basis Q. Define $P = Q \backslash \{l\}$ and the k-linear projection

$$\begin{aligned} k\{Q \cap \deg^{-1}(\deg(i) - 1)\} &\xrightarrow{\pi} k\{P \cap \deg^{-1}(\deg(i) - 1)\} \\ w &\mapsto w - \frac{w_l}{v_l} v \end{aligned}$$

where w_l and v_l are the coefficients of l in w and v expressed in the basis Q. For all $x \in F_{i-1}C_{\deg(i)-1}$ we define the function f as the composition $\pi \circ g$ and for all simplices $x \in F_{i-1}C_p$ with $p \neq \deg(i) - 1$, $f(x)$ is simply $g(x)$. Now we extend f to a k-linear map $f\colon F_iC \to k\{P\}$ by including $f(\{i\}) = 0$. Again by construction, f is a degree preserving projection with image $k\{P\}$. For $p \neq \deg(i)$ we have $F_{i-1}Z_p = Z_p$, and $Z_{\deg(i)} = F_{i-1}Z_{\deg(i)} \oplus k\{\{i\}\}$. Since $f(\{i\}) = 0$ and $\{i\} \notin Z_p$ we get $f(Z_p) = k\{P \cap \deg^{-1}(p)\}$ for all $p \in \mathbb{Z}$. Also $\ker(f) \cap Z_{p-1} = \operatorname{im}(\partial_p)$ for $p \neq \deg(i)$. For $p = \deg(i)$ we have

$$\begin{aligned} \ker(f) \cap Z_{\deg(i)-1} &= g^{-1}(k \cdot v) \cap Z_{\deg(i)-1} \\ &= (k \cdot \partial\{i\} + \ker(g)) \cap Z_{\deg(i)-1} \\ &= \operatorname{im}(\partial_{\deg(i)}). \end{aligned}$$

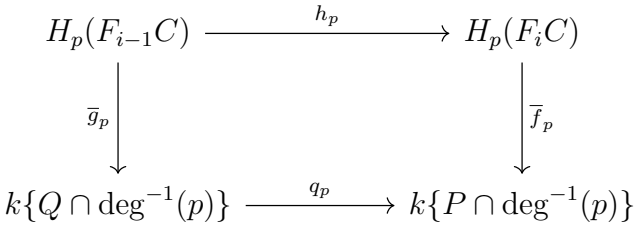
Thus, (P, f) is a homology basis for F_iC.

We know $P \subset Q$ so we define our homomorphism q to be

$$q(j) = \begin{cases} j & \text{if } j \in P \\ 0 & \text{if } j = l \end{cases}$$

for $j \in Q$.

Now we must show that the diagram in Definition 3.8 commutes, where the top homomorphism h is induced by the inclusion of $F_{i-1}C \subseteq F_iC$ and the bottom homomorphism is our chosen q. Both $\overline{g}$ and $\overline{f}$ are extensions of the isomorphisms as in Theorem 3.2. Both $\overline{g}$ and $\overline{f}$ can be restricted degree wise to isomorphisms $\overline{g}_p$ and $\overline{f}_p$ for $p \in \mathbb{Z}$ as every simplex contributes to the birth or death of a unique homology feature. The diagram in Definition 3.8 commutes only if the following degree wise restriction commutes for all $p \in \mathbb{Z}$.



Consider $p \neq \deg(i) - 1$. Then the diagram commutes as all maps are either inclusions or isomorphisms. Now consider $p = \deg(i)-1$ and $c \in H_{\deg(i)-1}(F_{n-1}C)$. If $c = 0$, $\overline{f}_{\deg(i)-1}(h_{\deg(i)-1}(c))$ and $q_{\deg(i)-1}(\overline{g}_{\deg(i)-1}(c))$ will also be zero. Assume $c \neq 0$ and has birth time t. If $t \neq l$, then the birth time of c is preserved under $h_{\deg(i)-1}$ as h_p is an inclusion map. If $t = l$, then both $\overline{f}_{\deg(i)-1}(h_{\deg(i)-1}(c))$ and $q_{\deg(i)-1}(\overline{g}_{\deg(i)-1}(c))$ will equal zero in $k\{P \cap \deg^{-1}(\deg(i)-1)\}$. Thus, the diagram commutes for all $p \in \mathbb{Z}$ and we have shown the diagram in Definition 3.8 commutes and that q is a homology basis contraction. □

We will use the prior two propositions in the proof of the following theorem.

Theorem 3.12. *Every totally filtered chain complex C has a minimal homology basis.*

Proof. We shall proceed inductively. Let $n = 0$, then the boundary homomorphism is zero and the statement holds trivially.

For our inductive step, assume (Q, g) is a minimal homology basis for $F_{n-1}C$ with $Q \subseteq [n-1]$ and the k-linear projection

$$g\colon F_{n-1}C \to k\{Q\}$$

Let $v = g(\partial\{n\})$. We may split this step into the $v = 0$ and $v \neq 0$ cases.

Case 1: $v = 0$.

Given that v is zero, we know that the simplex $\{n\}$ is a $\deg(n)$-cycle in $F_iC_{\deg(n)}$. This tells us that $H_*(F_{n-1}C) \subset H_*(F_nC)$. Hence, Proposition 3.10 induces a homology basis (P, f) of F_nC with a homology basis extension between $k\{Q\} \to k\{P\}$. The homology basis (P, f) was shown to be minimal in the proof of Proposition 3.10.

Case 2: $v \neq 0$.

If v is non-zero, we know that $\{n\}$ is a $\deg(n)$-boundary and that $H_*(F_nC) \subset H_*(F_{n-1}C)$. We can now utilize Proposition 3.11 to get both the homology basis (P, f) for F_iC and the homology basis contraction $q\colon k\{Q\} \to k\{P\}$.

Knowing the diagram

$$\begin{array}{ccc} H_*(F_{n-1}C) & \xrightarrow{\quad h \quad} & H_*(F_nC) \\ \Big\downarrow{\scriptstyle \overline{g}} & & \Big\downarrow{\scriptstyle \overline{f}} \\ k\{Q\} & \xrightarrow{\quad q \quad} & k\{P\} \end{array}$$

commutes, we now show that (P, f) is minimal. If $0 \neq c \in H_*(F_{n-1}C)$ and the birth time of c differs from the birth time l of the homology class represented by $\partial\{n\}$, then $h(c) \neq 0$ and the birth time of both is equal to the maximum element $i \in S$, the support of $\overline{g}(c)$. As the function q simply removes l from all supports in $k\{Q\}$ and $i \neq l$, we find that i is still the maximum element of the support $S/\{l\}$ of $q(\overline{g}(c))$. Thus, i is the maximum element of the support of $\overline{f}(h(c)) = q(\overline{g}(c))$ and we have a minimal basis. □

The proof of Theorem 3.12 serves as the genesis of the HomologyBasis algorithm presented in Section 4. The inductive step in cases 1 and 2 outline a method for the construction of what we call a *compatible sequence of homology bases* for a totally filtered chain complex based on $[n]$. Explicitly, compatibility in our context means the following.

Definition 3.13. Given a totally filtered chain complex C based on $[n]$, a sequence $\{(P_i, f_i)\}_{i=0}^n$ of homology bases is *compatible* if:

1. each P_i is the unique basis set induced by F_iC,
2. every homology basis (P_i, f_i) is minimal, and
3. each subsequent pair of homology bases (P_{i-1}, f_{i-1}) and (P_i, f_i) for $1 \leq i \leq n$ gives rise to a homomorphism q that is either a homology basis extension or a homology basis contraction.

3.2 Persistence in Homology Bases

Theorems 3.2 and 3.12 demonstrated how to extract the homology groups from a totally filtered chain complex using its homology basis. In doing so, we constructed sequences of homology bases that encode information on how the homological features evolve as the totally filtered chain complex is built. We now have enough to give a precise definition for the *persistent homology groups* of a totally filtered chain complex.

Definition 3.14. Given a totally filtered chain complex $C = (\partial, \deg)$ based on $[n]$, the p^{th} *persistent homology groups* are the images of the homomorphisms, $f_p^{i,j}\colon H_p(F_iC) \to H_p(F_jC)$ for $0 \leq i \leq j \leq n$ induced by the inclusion of $F_iC \hookrightarrow F_jC$.

As in the definition of persistent homology groups from Section 2.4, the homomorphism $f_p^{i,i}$ is the identity on $H_p(F_iC)$ for all i and p. We now have all the equipment required to compute persistent homology using the framework of our totally filtered chain complex C and the homology bases of its various F_iC restrictions.

Proposition 3.15. *Given a totally filtered chain complex C and a compatible sequence of its homology bases, the persistent homology groups of C can be computed.*

Proof. Let $C = (\partial, \deg)$ be a totally filtered chain complex based on $[n]$, with $\{(P_i, f_i)\}_{i=0}^n$ a compatible sequence of homology bases for C. Consider a homomorphism $f_p^{i,j}\colon H_p(F_iC) \to H_p(F_jC)$ induced by the inclusion $F_iC \hookrightarrow F_jC$. The homomorphism can be decomposed into a composition of homomorphisms

$$\{f_p^{k,k+1}\colon H_p(F_kC) \to H_p(F_{k+1}C)\}_{k=i}^{j-1}.$$

As either $H_p(F_kC) \subset H_p(F_{k+1}C)$ or $H_p(F_{k+1}C) \subset H_p(F_kC)$, each $f_p^{k,k+1}$ induces a homology basis extension or contraction $q_p^{k,k+1}\colon k\{P_k\} \to \{P_{k+1}\}$, by Propositions 3.10 and 3.11, which must live in our compatible sequence of homology bases. Given that $f_p^{i,j}$ is induced by the inclusion of F_iC in F_jC, when considering the image of $f_p^{i,j}$ in $H_p(F_jC)$ we ignore extensions and focus only on the homology basis contractions. The image of $f_p^{i,j}$ is then the image of P_i after applying all contractions between P_i and P_j. This is exactly one of the p^{th} persistent homology groups by Definition 3.14. After repeating this process for all p, i, and j we will have computed all the persistent homology groups of C. □

The definition of a homology class' *death time* in a totally filtered chain complex follows the intuition of Section 2.4.

Definition 3.16. Given a totally filtered chain complex $C = (\partial, \deg)$ based on $[n]$, the *death time* of a homology class $c \in H_pF_iC$ born at i, as described in Definition 3.6, is the minimal element $j > i$ of $[n]$ with the property that c is in the kernel of the homomorphism $H_p(F_iC) \to H_p(F_jC)$ induced by the inclusion $F_iC \subseteq F_jC$.

We say that a homology feature $c \in H_p(F_{i-1}C)$ with death time i *dies entering* F_iC.

Definition 3.17. Given a totally filtered chain complex $C = (\partial, \deg)$ based on $[n]$ and a homology class c that is born in F_iC and dies entering F_jC. The pair $(\{i\}, \{j\})$ is called the *persistence pair* for c. Any homology class that lacks a death time is called a *persistent feature.*

The use of brackets around i and j in the definition of persistence pairs is intentional to emphasize that a persistence pair of a totally filtered chain complex

is a tuple of elements in $[n]$, not of filtration values. As we have seen with birth times in a totally filtered chain complex, death times can also be converted to their corresponding filtration values in the finite filtered simplicial complex using the isomorphism $I: X \to [n]$. This means that when discussing birth times, death times, and persistence pairs, we may speak ambiguously as we can jump back and forth between (X, F) and C at will.

Example 3.18. Returning to the sequence of complexes in Figure 14, we wish to derive its persistence pairs. We have already determined the various homology bases for these for complexes: $P_2 = \{0, 1, 2\}$, $P_3 = \{0, 2\}$, $P_4 = \{0\}$, and $P_5 = \{0, 5\}$. This gives all the information required to compute the persistence pairs. First we notice that $\{0\}$ is never eliminated from our homology basis meaning it must be a persistent feature. However, $\{1\}$ and $\{2\}$ are removed from the basis at the contractions P_3 and P_4 respectively and thus give persistence pairs $(\{1\}, \{3\})$ and $(\{2\}, \{4\})$. Finally, the one dimensional homology feature born at $\{5\}$ dies entering $F_6\Delta^2$ if we extend the sequence to $F_6\Delta^2 = \Delta^2$ by adding the final 2-simplex. If we include the 2-simplex, we create the pair $(\{5\}, \{6\})$ Else we are left with two persistent features as seen in P_5.

Definition 3.19. Given a totally filtered chain complex $C = (\partial, \deg)$ based on $[n]$, a filtration function $F: [n] \to \mathbb{R}$, and homology class c with persistence pair $(\{i\}, \{j\})$, the *persistence* of c is defined as the real-value $F|_{\{j\}} - F|_{\{i\}}$.

Note that in the case that C represents some finite filtered simplicial complex (X, F), using the injective function $\Gamma_k: (X^*, F) \to \text{TFCC}$ from Proposition 2.30, then the filtration function used to compute persistence of homology features in C is $F \circ I^{-1}: [n] \to X \to \mathbb{R}$.

Theorem 3.20. *Every homology basis contraction induced by the inclusion $P \subset Q$, as described in Definition 3.8, corresponds to a unique persistence pair that dies entering F_iC.*

Proof. Let (P, f) be the homology basis formed by the contraction of (Q, g) as in the proof of Theorem 3.12. By construction, $g(\partial\{i\}) \neq 0$ and there exists a simplex $\{l\}$ in $F_{i-1}C$ such that $\{l\} = \max(\text{support}(\partial(\{i\})) \cap Q)$ and $\{l\} \notin Q$. Let $\deg(l) = k = \deg^{-1}(i) - 1$. As $\{l\} \in Q$, $\{l\}$ is the representative of a homology feature c of $H_k(F_{i-1}C)$. Also the homology basis for F_lC is the earliest basis $\{l\}$ could exist in. Hence, l is the minimal element of $[n]$ such that $\{l\}$ is in the image of the homomorphism $H_{\deg^{-1}(i)-1}(F_lC) \to H_{\deg^{-1}(i)-1}(F_{i-1}C)$ induced by the inclusion $F_lC \subseteq F_{i-1}C$. Then by Definition 3.6, the birth time of c is l. As $\{l\} \in P$ but $\{l\} \notin Q$, we find that F_iC is the the earliest totally filtered chain complex that the homology class c, represented by $\{l\}$, is in the kernel of the homomorphism

$H_{\deg^{-1}(i)-1}(F_lC) \to H_{\deg^{-1}(i)-1}(F_iC)$. By Definition 3.16, the death time of c is then i. We have now formed the persistence pair $(\{l\}, \{i\})$.

Uniqueness of the persistence pairs follows from the uniqueness of each simplex that induced the homology basis contractions and extensions in our compatible sequence of homology bases for C. Hence, each persistence pair uniquely represents a homological feature that is born in F_lC and dies entering F_iC. □

4 The HomologyBasis Algorithm

We now present a general, theoretically explicit, schema for the computation of a totally filtered chain complexes homology basis and persistent homlogy, in the form of its corresponding persistence pairs. The algorithm itself may be found here: HomologyBasis

4.1 General Construction

As stated, our goal is to construct an efficient way of computing a finite filtered simplicial complex's persistent homology, namely the set of persistence pairs. Starting with a given finite filtered simplicial complex, we can generate a totally filtered chain complex C by choosing a field k and applying Γ_k from Proposition 2.30. This allows us to utilise Theorem 3.12 to construct a compatible sequence of homology bases. During the sequences construction we apply Theorem 3.20 when appropriate to determine all finite persistence pairs. Lastly, we account for the remaining persistent feature of our chain complex and add them to our set of persistence pairs.

We now expand precisely on this schema. Given a totally filtered chain complex $C = (\deg, \partial)$, we begin building a compatible sequence of homology bases from the minimal homology basis $(P_0 = \{0\}, f_0)$ with $f_0 = * \circ \mathbf{f}_0$ being the map that takes all functions $\psi\colon [n] \to k$ to the composition $\psi \circ \mathbf{f}_0$ where $\mathbf{f}_0$ is the basis element of C generated by $\{0\}$.

Any transition from (P_{i-1}, f_{i-1}) to (P_i, f_i) for $i = 1, \ldots, n$ will be characterized by a homology basis extension or contraction depending on if a new cycle is created by the addition of the simplex $\{i\}$. In the case of extension, the addition of $\{i\}$ creates a new cycle in the totally filtered chain complex F_iC that was not present in $F_{i-1}C$ telling us that the image of $\partial\{i\}$ under f_{i-1} is zero. Then we have a new homology feature with birth time equal to i and we choose the extension of (P_{i-1}, f_{i-1}) to (P_i, f_i) where $P_i = P_{i-1} \cup \{i\}$ and f_i is simply f_{i-1} with $\{i\}$ mapped to itself.

If $\{i\}$ does not form a new cycle, we know we are dealing with a homology basis contraction. Let $\{i\}$ be the new d-simplex added to F_iC. We know $\partial\{i\}$ is non-zero and $\{i\}$'s addition necessitates the destruction of some homological feature from $H_*(F_{i-1}C)$. Let $\{l\}$ represent the maximum element in the support of $f_{i-1}(\partial\{i\})$ with respect to the prior homology basis. The simplex $\{l\}$ is the youngest simplex in the boundary of $\{i\}$'s projection into P_{i-1} and, as $l \in P_{i-1}$, the Elder Rule dictates that the the homology feature l must die entering F_iC. This is represented in our homology basis by choosing $P_i = P_{i-1} \backslash \{l\}$ to be the subset of $[n]$ our basis will be contracted onto. Now define the projection f_i as we

did in Theorem 3.12. Using the k-linear projection π we choose

$$f_i(\sigma) = \begin{cases} f_{i-1}(\sigma) & \text{if } \sigma \in F_{i-1}C_p \text{ for } p \neq d-1 \\ \pi(f_{i-1}(\sigma)) & \text{if } \sigma \in F_iC_{d-1} \\ 0 & \text{if } \sigma = \{i\}. \end{cases}$$

This contracts (P_{i-1}, f_{i-1}) to (P_i, f_i) and as we proved in Theorem 3.20, generates the new persistence pair $(\{l\}, \{i\})$.

Applying the extension and contraction cases when proper allows us to iterate through all $i \in [n]$ and construct both the homology bases in the compatible sequence $\{(P_i, f_i)\}_{i=0}^n$. As we have only recorded the persistence pairs of finite persistence, we are left to account for the persistent features of C. Notice that each element of P_n represents a homology feature, there are no more $F_{i>n}C$ to contract to, and each feature in P_n never die entering one of the totally filtered chain complexes F_iC for $0 \leq i \leq n$. Hence, each is a persistent feature of C and we find the persistence pair $(\{k\}, \infty)$ for every $k \in P_n$. Thus, our proposed schema finds both our homology basis for $F_nC = C$ and all its persistence pairs.

4.2 HomologyBasis

The schema described in the prior Section 4.1 is now formalized into the algorithm *HomologyBasis* illustrated in Algorithm 1. The two sub algorithms 2 and 3 are described in further detail in the subsequent subsections.

Expanding upon Algorithm 1. Let (X, F) be a filtered simplicial complex with injective filtration function $F\colon X \to \mathbb{N}$ such that given $\sigma, \tau \in X$ if $\sigma \subseteq \tau$ then $F(\sigma) \leq F(\tau)$. This will be the complex that we compute persistent homology for, in a chosen field k. We convert (X, F) into a totally filtered chain complex $C = (\partial, \deg)$ as in Proposition 2.30 and begin to build our desired compatible sequence of homology bases. Restricting to F_0C allows us to start the compatible sequence as any totally filtered chain complex has a minimal homology basis for F_0C of $(P_0 = \{0\}, f_0 = * \circ \mathbf{f}_0)$.

For $1 \leq i \leq n$, we either extend or contract the $F_{i-1}C$ homology basis depending on if $\{i\}$ is a cycle in C. This is accomplished, as in Theorem 3.12, by applying Algorithm 2 when $\{i\}$ is a cycle and Algorithm 3 when it is not. Both of these algorithms are explored more thoroughly later in this section.

By Theorem 3.20, we can uniquely determine all finite persistence pairs of C from exclusively the homology basis contractions in the compatible sequence of homology bases. Every contraction will remove a simplex from the homology basis. If $\{l\}$ is the removed simplex, then it is the representative for the homological feature that is born at time l and dies at time i. Thus, we create the persistence

Algorithm 1 HomologyBasis.
Input: finite filtered simplicial complex (X, F) and a field k
Output: Homology basis (P, f) and the persistence pairs

Convert (X, f) to $C = (\partial, \deg)$
$P = \{0\}$
$f =$ pre-composition with $p_{\{0\}}\colon [n] \to \{0\}$
persistence pairs $= \emptyset$
for $i = 1, \dots n$ **do**
  if $f(\partial\{i\}) = 0$ **then**
    $(P, f) =$ HBE(C,P,f)
  else
    $(P, f) =$ HBC(C,P,f)
    add $(\{l\}, \{i\})$ to persistence pairs
  end if
end for
for $\sigma \in P$ **do**
  add (σ, ∞) to persistence pairs
end for
return (P, f) and persistence pairs

Algorithm 2 HBE (Homology Basis Extension).
Input: Totally filtered chain complex $C = (\partial, \deg)$ and homology basis (P, f)
Output: Homology basis (Q, g)

$Q = P \cup \{i\}$
$g = f$ extended to include $\{i\} \mapsto \{i\}$
return (Q, g)

pair $(\{l\}, \{i\})$ every time we find a homology basis contraction between $k\{P_{i-1}\}$ and $k\{P_i\}$.

Once we have preformed the above operations for all $i = 1, \dots, n$. We will have all the information of the persistence pairs we need. To find the persistent features of C, we look at the final homology basis (P_n, f_n). All elements in P_n represent homology classes that persist up to and through $F_nC = C$, making P_n the collection of persistent features. To represent these persistent features as persistence pairs we create the pair $(\{k\}, \infty)$ for every $k \in P_n$. We now have the desired persistence pairs and have extracted all persistence information from our finite filtered simplicial complex.

Algorithm 3 HBC (Homology Basis Contraction).
Input: Totally filtered chain complex $C = (\partial, \deg)$ and homology basis (P, f)
Output: Homology basis (Q, g)

$\{l\} = \max(\text{support}(f(\partial\{i\})))$, $d = \deg(i)$
$Q = P/\{l\}$
$g = f$
for $\sigma \in k\{P \cap \deg^{-1}(d-1)\}$ **do**
  $g(\sigma) = \pi \circ g(\sigma)$ with π from Proposition 3.11
end for
extended g to include $\{i\} \mapsto 0$
return (Q, g)

4.2.1 Extension

Given the homology basis (P_{i-1}, f_{i-1}) for $F_{i-1}C$, assume that the new simplex has a zero boundary in the $F_{i-1}C$ homology basis. Then $\{i\}$ will be a cycle of F_iC and $H_*(F_{i-1}C) \subset H_*(F_iC)$. We wish to induce a homology basis extension between $k\{P_{i-1}\}$ and $k\{P_i\}$. This is accomplished by the choice of P_i and f_i as in Algorithm 2 where we extend P_{i-1} to $P_{i-1} \cup \{i\}$ and extend the f_{i-1} to include $\{i\} \mapsto \{i\}$.

The homology basis extension does give us the the birth time information of $c \in H_*(F_iC)$, the homology class generated by the addition of $\{i\}$. However, it gives us no insight into whether c is of finite or infinite persistence. Thus, we leave the determination of this and the construction of any persistence pairs to the next contraction algorithm.

4.2.2 Contraction

Given $\{i\}$ is not a cycle in F_iC, so we know that it destroys some homology feature c characterized by $\partial\{i\}$. In the homology basis (P_{i-1}, f_{i-1}), this homology feature is represented by the the simplex that causes its birth, namely the simplex $\{l\} \in \partial\{i\}$ that was last added to the complex. As we add simplices to our complex in order of filtration value, $\{l\}$ is precisely the maximum element in $\partial\{i\}$'s representation under the prior basis (P_{i-1}, f_{i-1}): $\max(\text{support}(f_{i-1}(\partial\{i\})))$. Thus, we know precisely which element of P_{i-1} will be removed to get the correct P_i.

Now we must contract the k-linear projection f_{i-1} in such a way that we induce the desired homology basis contraction. If $\{i\}$ has a degree of d, the only basis elements for $k\{P_{i-1}\}$ that can have $\{l\}$ in their support are those that correspond to $(d-1)$-simplices in $F_{i-1}C$ and hence, the projection f_{i-1} will be unchanged on the domain $k\{P_{i-1} \cap \deg^{-1}(d' \neq d-1)\}$. Letting $v = f_{i-1}(\partial\{i\})$, choose a change

of basis

$$\pi\colon k\{P_{i-1} \cap \deg^{-1}(d-1)\} \to k\{P_i \cap \deg^{-1}(d-1)\}$$

defined by $w \mapsto w - (\frac{w_l}{v_l})v$, where w_l and v_l are the coefficients of $\{l\}$ in w and v respectively. Applying the composition $\pi \circ f_{i-1}$ then removes all trace of $\{l\}$ from the prior homology basis. Combining the two cases above, we extend the projection into $k\{P_{i-1}\}$ using the map $\{i\} \mapsto 0$. This is exactly the k-linear projection that induces a homology basis contraction between $k\{P_{i-1}\}$ and $k\{P_i = P_{i-1}/\{l\}\}$.

As discussed in Theorem 3.20, every contraction will create the finite persistence pair $(\{l\}, \{i\})$. This is utilized in Algorithm 1 to find all finite persistence pairs for our totally filtered chain complex.

Remark. *A notebook giving a bare bones implementation of the above pseudo code is provided here: HomologyBasis-Implementation.*

5 Simplex Tree

To determine the computational efficiency of HomologyBasis, we compare to well established algorithm GUHDI. For the storage of abstract simplicial complexes, GUHDI uses a tree data structure called *Simplex Tree*. We now give a brief overview of simplex tree. Introduced by Boissonnat and Maria [3], it aims to efficiently represent abstract simplicial complexes of any dimension. This is accomplished by constructing a bijective correspondence between the faces of a given simplicial complex and the nodes of a trie as follows.

Definition 5.1. Given a simplicial complex (X, V) of dimension k with vertices enumerated 1 up to $\|V\|$. Then each simplex in X can be associated to a unique word on the alphabet $1 \dots \|V\|$ and a *simplex tree* structure is a trie on the words representing simplices in (X, V) such that:

1. The nodes of the simplex tree are in bijective correspondence with the simplices, of all dimensions, of the complex (X, V). Note the root is associated to the empty simplex.

2. Each node of the tree, except the root, stores the label of a vertex. Specifically the label of the maximum enumerated vertex in the subset $\sigma \in X$.

3. Traveling along a path from the root gives a simplex whose vertices are the vertex labels encountered on the path and the labels are always encountered in increasing order.

For an explicit example, we will represent the $\check{\mathrm{C}}_4(S)$ complex from Figure 6. Uniquely associating each vertex from this complex with a number from $\{1, \dots, 8\}$ as seen in Figure 15.

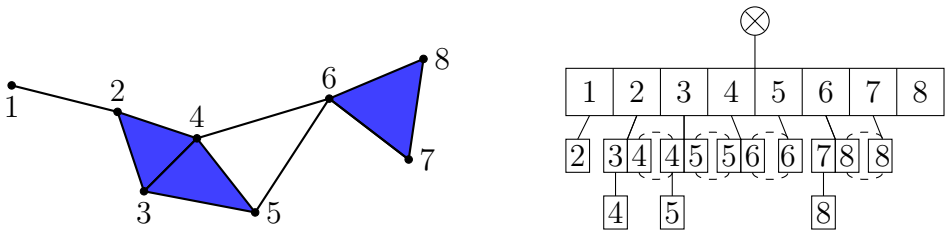


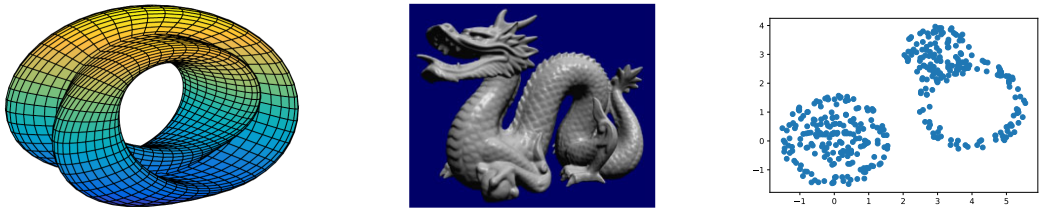
Figure 15: The $\check{\mathrm{C}}_4(S)$ complex on 8 vertices and its representation as a SimplexTree trie.

As per the definition, the root represents the empty face and serves as the boundary to all 0-simplices. Notice that the depth of each node in the SimplexTrere is equal to the dimension of the simplex that the node represents plus one. Thus,

as we travel deeper into the SimpleTree we find simplices of higher and higher dimension. All boundary relations can be determined from the trie by starting at the given node, moving back toward the root on level, and identifying all linked simplex nodes at this depth. Take the word $[2, 3, 4]$ that describes the 2-simplex $[2, 3, 4]$ in $\check{\mathrm{C}}_4(S)$. Moving back toward the root we find the words $[2, 3]$ and $[2, 4]$. Both of these simplices are boundary elements of our simplex, however there is still one unaccounted for boundary term. In order to find this last boundary face, we must traverse the circular list (represented by a dotted loop) connecting the words $[2, 4]$ and $[3, 4]$. The boundary faces of the simplex $[2, 3, 4]$ are $[2, 3]$, $[2, 4]$, and $[3, 4]$, which are exactly the nodes we have picked out in the SimplexTree. The use of circular lists also allows us to quickly locate all instances of a given label in the tree. For additional information on the efficiency of tries see [2].

6 Results

All comparisons between GUDHI and our proposed HomologyBasis algorithm, were conducted on a laptop with an Intel Core i7-7820HQ CPU processor running at 2.90Ghz and 8.00 GB of installed RAM. The operating system was 64-bit Windows 10 pro version 20H2. The data sets used as benchmarks for the comparisons were a 900 point sampling of the *Klein Bottle* in its figure-8 embedding in $\mathbb{R}^3$, a 2000 point sampling of the *Stanford Dragon*, and a 5000 point sampling of the 5-ball of radius 1 centered at the origin. These may be seen in Figures 16a and 16b. For reasons that will be discussed in the next Section 6.5, we have limited most comparisons in this thesis to simplicial complexes of degree no larger than 3. Additionally, the data set *Bull's Eye*, consisting of the union of 400 randomly generated points on the 1-sphere centered at the origin with radius 1 and 200 randomly generated points in the annulus centered at the origin with inner radius 1.25 and outer radius 1.5, see Figure 16c, is used as an explicit verification of the agreement between the two algorithms. All resulting times for the various comparisons may be found here: Comparison-Data-GUHDI-vs-HomologyBasis.



(a) Klein Bottle (b) Stanford Dragon (c) Bull's Eye

Figure 16: From left to right we have the figure-8 embedding of the Klein Bottle, the Stanford Dragon, and the Bull's Eye distribution. From the former two, 900 and 2000 data points are sampled respectively.

Before proceeding, we introduce the term *critical complex* to be a Euclidean data complex upon which both GUHDI and HomologyBasis compute persistent homology in roughly the same amount of time and after which, one becomes consistently more efficient than the other. Note that whenever a critical complex is specified, it is a rough approximation of where the critical complex occurs.

The focus of our analysis is two fold. First and foremost, we wished to confirm that both algorithms are in agreement when computing persistent homology and persistence pairs. This was accomplished by a checking step added to every computation to ensure both GUHDI and HomologyBasis produced the same persistence pairs. This verification step is not included in the computation time for the either algorithm and as expected both were in agreement for all simplicial

complexes tested. For an explicit example of this see Section 6.1. The second goal of our analysis is to determine the computational strengths of our proposed HomologyBasis algorithm with respect to that of GUHDI. Seeing as both algorithms "forget" the position of vertices, only storing dimension and boundary relationships for each simplex, it was considered unlikely that the dimension of the ambient space from which the point clouds were sampled would have a significant effect on computational efficiency. This was verified by computing the persistent homology for the complete 1-skeletons of two 2000 point sampling from the 3 and 15-ball respectively. Even though both of the vertex sets of the simplicial complexes were randomly generated, as both were complete 1-skeletons they claimed membership of the same isomorphism class and would be represented identically by HomologyBasis. A similar argument can be made for the complexes' representations by SimplexTree. There was no significant difference in computation times for either sampling.

6.1 Bull's Eye

We now present the qualitative results to a comparison of the persistence pairs computed by both GUHDI and HomologyBasis. Specifically, computing the persistence pairs of the complete 2-skeleton with the Bull's Eye point cloud acting as the vertex set. This has been added as an example to the HomologyBasis repository here: HomologyBasis-GUHDI-Homology-Check. The persistence pairs computed by both algorithms are presented in persistence and barcode diagrams in Figure 17. The diagrams serve as an easy eye test for verifying that HomologyBaiss is producing the same persistent homology as GUHDI. It is important to note that the persistence and barcode diagrams are identical for both algorithms. We find there is one persistent 0-homology feature, as well as a spike in longer lasting 1-homology features being born around the filtration value 0.15, generated by the vertices sampled from the annulus.

6.2 Klien Bottle

All the remaining data sets were used to compare computational run times of the two algorithms. The first we will discuss is that of the figure-8 embedding of the Klein bottle in $\mathbb{R}^2$. In order to best understand how the run time of HomologyBasis scales as more simplices are added to the simplicial complex, the number of points used to construct our Euclidian data complex was varied from 50 to 900. The Euclidian data complex constructed on each sampling were the complete k-skeletons for $k \in \{1, 2, 3\}$. Only k-skeletons of the same dimension are compared with each other and the maximum number of sampled vertices for $k = 2$ and 3

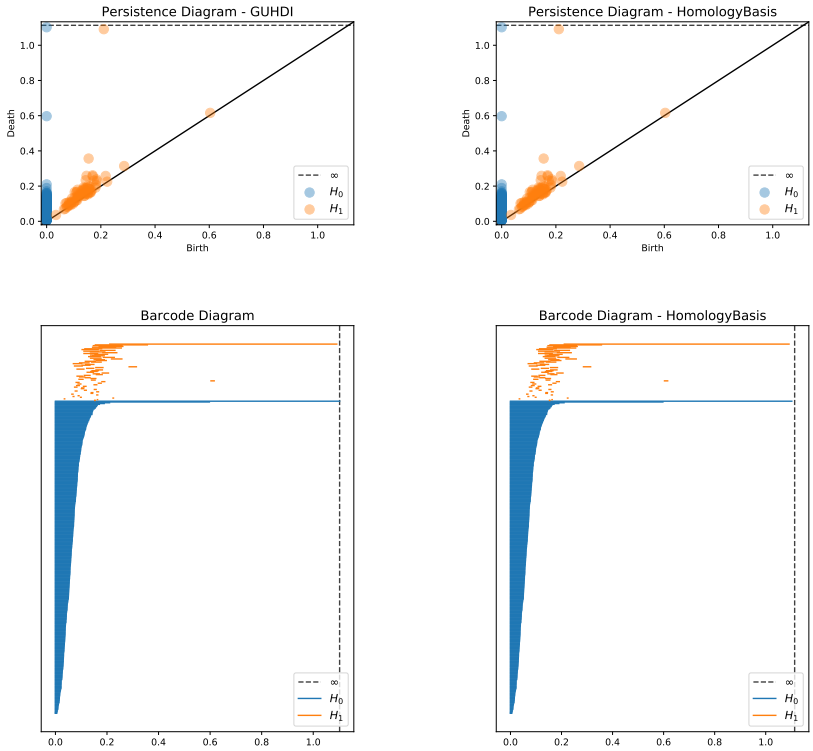


Figure 17: The persistence and barcode diagrams for the complex 2-skeleton complex constructed on the Bull's Eye point cloud. Each diagram is labeled by the algorithm used to compute the persistence pairs it presents.

were restricted to 400 and 180 respectively. We will proceed by analyzing each dimension on its own.

For $k = 1$, when only looking at the raw run times, GUHDI is consistently faster. However, as can be seen in Figure 18, when focusing on the proportional computation times we find a steady decrease as more and more 1-simplices are added to the Euclidian data complex. This leads us to believe that there may exist a critical complex, after which HomologyBasis becomes superior for computing persistent homology, but the total number of points sampled is not enough to construct it. This is further supported by the findings in both Sections 6.3 and 6.4.

The critical complexes are clearly identifiable for both the 2 and 3-skeletons, occurring at roughly 110 and 60 sampled points respectively. As seen in Figure 19, the decreasing proportional run time exhibited in the 1-skeleton comparisons is still visible when $k = 2$ and 3. However, the incremental decrease seems to

approach 0 as the proportional run times approach $1:2$.

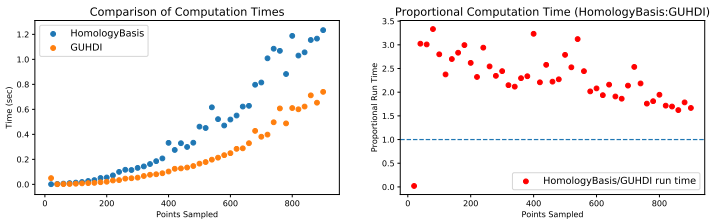


Figure 18: Computational efficiency comparison between GUHDI and HomologyBasis on various complete 1-complexes sampled from the embedded Klein Bottle data set. Note the lack of a critical complex.

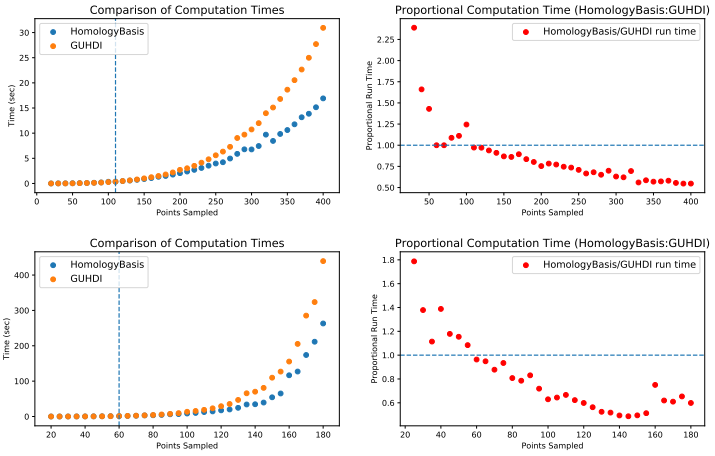


Figure 19: The computational efficiency comparisons between GUHDI and HomologyBasis on various complete 2-complexes (above) and 3-complexes (below) sampled from the embedded Klein Bottle data set. Note the critical complexes occur around $n = 110$ and $n = 60$ respectively.

6.3 Stanford Dragon

Similar to the comparisons done in Section 6.2, we split our analysis into the complete 1, 2, and 3-skeletons with the number of points used to generate the Euclidian data complex being our variable of interest. The 3-skeleton is presented in the notebook GUHDI-vs-HomologyBasis-Speed-Comparison. As the Stanford Dragon point cloud is over twice the size of the Klein Bottle, we are able to get a better picture of the run time behaviour for both algorithms when applied to larger and larger simplicial complexes. This is most noticeable for the complete

1-skeleton, see in Figure 20, as a critical complex does appear as the vertex set of our Euclidian data complex reaches 1900 points. However, the critical complex occurs so close to the total sampling of 2000 points that it limits our ability to see what happens after the critical complex. As the same proportional decrease with respect to number of sampled vertices is observed as in Section 6.2, it is assumed that HomologyBasis is either comparable to GUHDI or better for complete 1-skeletons of 2000 points or more.

Figure 21 shows the Stanford Dragon comparisons for the complete 2 and 3-skeletons. As in the Klein Bottle comparison, we find the critical complexes very quickly, 130 and 55 points respectively. One difference between the two data sets comparisons is that the proportional computation time for the 3-skeleton drops below the $1:2$ mark in this comparison where as it did not in Section 6.2. It is currently unknown if there is a limiting proportional run time for the two algorithms.

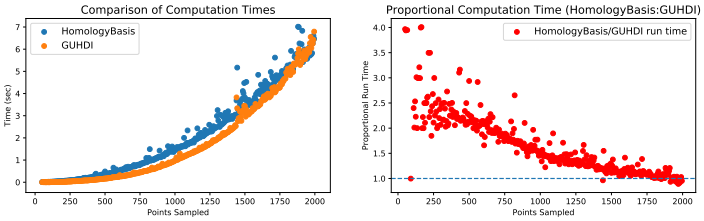


Figure 20: Computational efficiency comparison between GUHDI and HomologyBasis on various complete 1-complexes sampled from the Stanford Dragon data set. Notice how the computational times trend towards one another as the complexes grow larger.

6.4 5-ball

In the 1-skeleton comparison of the prior two data sets we were not able to adequately capture long term run time differences between GUHDI and HomologyBasis due to the limited size of each point cloud. This next example is wholly focused on determining the behaviour of each algorithm after the proposed critical complex from Section 6.3. To do this a much larger point cloud was generated consisting of 5000 distinct vertices laying within the 5-ball and computation times were compared for various complete 1-complexes. The results of this comparison are presented in Figure 22. As expected, we find a critical complex around the 2000 point sampling such that afterwards the HomologyBasis algorithm consistently outperforms GUHDI. It seems from this comparison that the proportional computation time is not determined by a linear relationship and, as we apply the

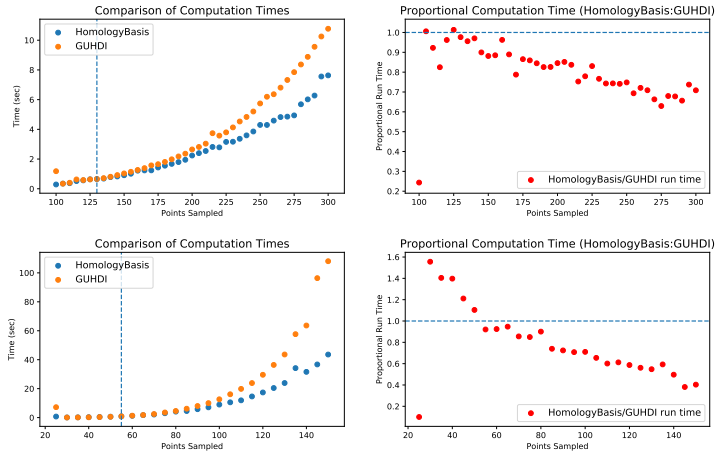


Figure 21: The computational efficiency comparisons between GUHDI and HomologyBasis on various complete 2-complexes (above) and 3-complexes (below) sampled from the Stanford Dragon data set. Notice both he inverse relation between the size of the complex and the proportional run time of the two algorithms and that, as the simplicial complexes grow larger, HomologyBasis overtakes GUHDI in terms of computational efficiency.

algorithms to larger and larger point clouds, we approaching a limiting proportional difference between the two algorithms run times.

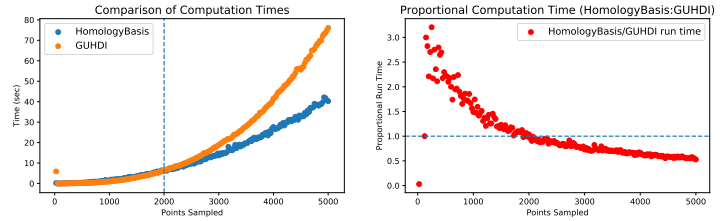


Figure 22: Run time comparison between GUHDI and HomologyBasis on various complete 1-complexes sampled from 5-dimensional ball of radius 1 centered at the origin.

6.5 Final Remarks

Now we address a major draw back of the HomologyBasis algorithm in its current form. That problem is computational efficiency when simplices of dimension 4 and greater are included in the Euclidian data complex. This issue was noticed after computing the persistent homology for a complete 4-complex sampled from

the Stanford Dragon data set and was followed up on by computing multiple complete 4 and 5-complexes sampled from a 3-ball. The results of both can be seen in Figure 23. The most stark contrast to the comparisons in prior dimensions, is how HomologyBasis quickly becomes orders of magnitude slower than GUHDI when 4 and 5-simplices are introduced. The rapid explosion in run time occurs when the Euclidian data complexes are constructed on 40 and 20 vertices respectively. Given that there are only 7.6×10^5 simplices in a complete 4-complex of 40 vertices and 6.0×10^4 in a complete 5-complex of 20 vertices, the driving factor behind HomologyBasis' computational inefficiency seems to be the dimension of the complex rather than its size. It is believed that this issue may be remedied

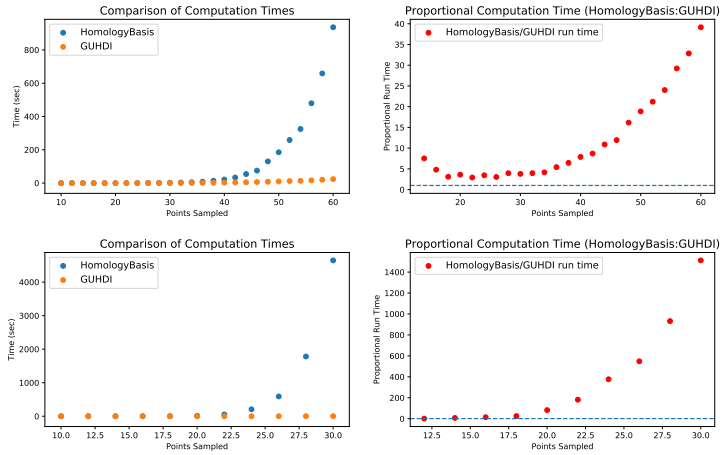


Figure 23: Run times and proportional run times for various complete 4-complexes (above) and complete 5-complexes (below) by the GUHDI and HomologyBasis algorithms. Note the extremely rapid increase of the HomologyBasis run time.

through the implementation of HomologyBasis in C++ or through a more efficient implementation of the algorithm. These adjustments are beyond the scope of this thesis and are left as future alterations to HomologyBasis.

In an effort to allow for easy transition from GUHDI to HomologyBasis, the author constructed a class within HomologyBasis that replicates the functionality of GUHDI's SimplexTree. This class goes by the same name but uses HomologyBasis' techniques for computing persistent homology. The goal was to take code that uses GUHDI to compute persistent homology for a finite filtered simplicial complex, replace GUHDI with HomologyBasis when importing packages, and increase the efficiency without altering any of the code itself. An implementation of this is seen in Figure 24. As we have consistently found critical complexes around the $n = 2000$ mark for complete 1-complexes, after which HomologyBasis more efficiently computes persistent homology, it is believed that the inefficiency

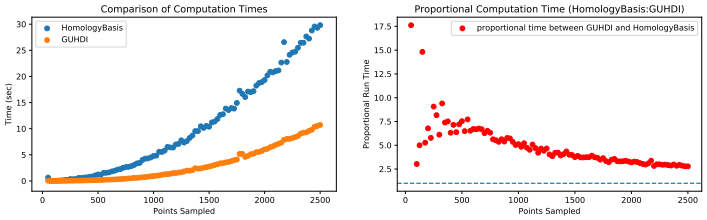


Figure 24: Comparison between GUHDI's SimplexTree and HomologyBasis' SimplexTree classes' run time on the complete 1-complex constructed on a sampling of 2500 points from the 2-ball with radius 1 centered at the origin.

demonstrated in Figure 24 is a result of a less than optimized implementation of HomologyBasis' SimplexTree class. This comparison is presented in the notebook: GUHDI-vs-HomologyBasis-SimplexTree.ipynb

7 Further Research

As seen in Section 6.5, HomologyBasis experiences a sharp drop in efficiency when simplices of dimension greater than 3 are introduced. This is not believed to be due to a failure in the practical implementation of HomologyBasis rather than in the foundation upon which the algorithm is built. It is of interest to the author to further explore more streamlined and efficient implementations of the HomologyBasis algorithm in order to see if its usefulness does indeed extend beyond the scope of simplicial complexes of dimension 3, as is believed.

The goal of HomologyBasis is to form a code that can be used to directly replace that of GUHDI. In order to do this, it is required to have a class that is functionally identical to GUHDI's SimplexTree. We have seen that while HomologyBasis has a rough implementation of this, it has not been done optimally and thus, hinders the advantages that HomologyBasis holds over GUHDI. It is of interest to explore more optimal implementation of this class, with the end goal of being able to implement HomologyBasis in any existing code that uses GUHDI simply by changing one line.

Additionally, while the author is not well versed in the C++ language, an exploration into the comparative efficiencies of GUHDI and a C++ implementation of HomologyBasis would serve as a accurate test between the two. Given that code in C++ can always be more optimized than in pure Python, it is believed that HomologyBasis would only perform better on this leveled playing field.

Lastly, while we have a category theoretic description of finite filtered simplicial complexes, we are yet to apply the same terminology to totally filtered chain complexes. Lacking this formal definition bars us from describing the injective function Γ_k from Proposition 2.30 as a functor. As this was the initial goal of the author, there is still work to be done in framing the set totally filtered chain complexes as a category.

8 Conclusion

In this thesis, we presented the totally filtered chain complex and its respective homology basis as a theoretic foundation for the HomologyBasis algorithm. HomologyBasis proved to utilize this foundation to efficiently compute the persistent homology of large low-dimensional simplicial complexes.

In the various comparisons of Section 6, we saw that when limited to simplices of dimension 0, 1, 2, and 3, the more simplices in our complex the better HomologyBasis performed compared to GUHDI. During the best comparisons, HomologyBasis was able to produce persistence pairs at approximately double the efficiency of GUHDI. It is still unknown if there is a limit to the proportional efficiency of these comparisons. However, HomologyBasis' efficiency falters when investigating simplicial complexes of dimension 4 and greater as seen in Section 6.5. It should be remembered the difference in implementations between the two algorithms, GUHDI in C++ and HomologyBasis in pure Python, which will be a contributing factor but is not believed to be the main cause of this stark difference in run times between dimensions. The lack of efficiency is believed to stems from less than optimal implementation of the HomologyBasis algorithm and that this is a problem that can be remedied through the use of more creative coding. However, this has not been verified.

Ultimately, HomologyBasis presents itself as a very intriguing alternative to one of the more popular algorithms for the computation of persistent homology. While it is by no means a finished product and has a less than optimal implementation, it shows strong indications that it is an algorithm that can compete, and even surpass, the benchmark persistent homology algorithms of the present day. This alone makes it worth further investigation.

References

[1] Aaron Adcock, Daniel L. Rubin, and Gunnar Carlsson. Classification of hepatic lesions using the matching metric. *CoRR*, abs/1210.0866, 2012.

[2] Jon Louis Bentley and Robert Sedgewick. Fast algorithms for sorting and searching strings. *SODA*, pages 360–369, 1997.

[3] Jean-Daniel Boissonnat and Clément Maria. The simplex tree: an efficient data structure for general simplicial complexes. *CoRR*, abs/2001.02581, 2020.

[4] Erin Chambers, Vin Silva, Jeff Erickson, and Robert Ghrist. Vietoris–rips complexes of planar point sets. *Discrete & Computational Geometry*, 44:75–90, 07 2010.

[5] Joseph Minhow Chan, Gunnar Carlsson, and Raul Rabadan. Topology of viral evolution. *Proceedings of the National Academy of Sciences*, 2013.

[6] Chad Giusti, Robert Ghrist, and Danielle S. Bassett. Two's company, three (or more) is a simplex. *Journal of Computational Neuroscience*, 41(1):1–14, Aug 2016.

[7] Allen Hatcher. *Algebraic topology.* Cambridge University Press, 2002.

[8] John L. Harer Herbert Edelsbrunner. *Computational Topology: An Introduction.* American Mathematical Society, 2009.

[9] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. *Topological Persistence and Simplification.* Discrete & Computational Geometry 28, 2002.

[10] Huang-Wei Chang, Sergio Bacallado, Vijay Pande, and Gunnar Carlsson. Persistent topology and metastable state in conformation dynamics. *PLoS ONE*, 8(4): e58699, 2013.

[11] Monica Nicolau, Arnold J. Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, 2011.

[12] Nina Otter, Mason A. Porter, Ulrike Tillmann et al. A roadmap for the computation of persistent homology. *EPJ Data Sci.*, 6(17), 2017.

[13] Achille C. Varzi. The magic of holes. In Pina Marsico and Luca Tateo, editors, *Ordinary Things and Their Extraordinary Meanings*, pages 21–33. Charlotte (NC): Information Age Publishing, 2019.